UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Elétrica e de Computação

SOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO LINEAR POR REDES NEURAIS ASSOCIADAS A MÉTODOS DE PONTOS INTERIORES.

Marta Ines Velazco Fontova
Orientador: Prof. Dr. Christiano Lyra Filho
Co-Orientador: Aurelio Ribeiro Leite de Oliveira

Banca Examinadora:

Christiano Lyra Filho –DENSIS/FEEC/UNICAMP

Ivan Nunes da Silva –FEB/UNESP

Alexandre Pinto Alves da Silva –COPPE/UFRJ

Vinícius Amaral Armentano -DCA/FEEC/UNICAMP

Wagner Caradori do Amaral -DCA/FEEC/UNICAMP

Fernando José Von Zuben -DCA/FEEC/UNICAMP

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para obtenção de título do Doutor em Engenharia Elétrica.

Campinas, São Paulo.
Dezembro 2003

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Velazco Fontova, Marta Ines
V541s Solução de problemas de otimização linear por redes neurais associadas a métodos de pontos interiores / Marta Ines Velazco Fontova.--Campinas, SP: [s.n.], 2003.

Orientadores: Christiano Lyra Filho e Aurelio Ribeiro Leite de Oliveira.
Tese (Doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes neurais (Computação). 2. Programação linear. 3. Otimização matemática. I. Lyra Filho, Christiano. II. Oliveira, Aurelio Ribeiro Leite de. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

RESUMO

O trabalho investiga alternativas para solução de problemas lineares de otimização através da cooperação entre redes neurais e métodos de pontos interiores, procurando identificar benefícios na fertilização cruzada entre essas alternativas. A análise do conjunto de contribuições anteriores sobre aplicação de conceitos de redes neurais à solução de problemas de otimização sugere que houve pouca troca de informações entre essas áreas—redes neurais e otimização. Em particular, todas as abordagens exploradas foram aplicadas a exemplos ilustrativos pequenos, que poderiam ser resolvidos sem auxílio de qualquer recurso computacional; estão muito distantes dos problemas reais de grande porte abordados por pesquisadores da área de otimização. Neste trabalho, redes neurais de Hopfield são utilizadas nas etapas iniciais de solução dos problemas de otimização, passando as informações parciais para os métodos de pontos interiores, que concluem o processo de solução. Investigam-se alternativas de cooperação entre redes neurais de Hopfield e as principais famílias de métodos de pontos interiores: métodos afins escala e métodos primais-duais. As metodologias propostas foram avaliadas em problemas reais do conjunto *Netlib*, permitindo extrair indicadores sobre a redução do número de iterações e do tempo total de processamento para obtenção de soluções ótimas.

ABSTRACT

This works explores possibilities of cooperation between neural networks and interior point methods to solve linear optimization problems. It seems that the neural networks and optimization communities carry on their research in worlds apart, with only tiny links between each other. Researchers in neural networks provided theoretical results for addressing optimization problems but did not go much beyond demonstrative examples; problems used to illustrate the optimization approaches by neural networks are small (most of them are textbook examples that can be solved by hand calculations). In this work, Hopfield neural networks and interior point methods are used in an integrated way to solve linear optimization problems. Hopfield networks perform the early stages of the optimization procedures, giving enhanced feasible starting points for interior point methods, which can be way ahead in the path to optimality. Cooperation with both the affine scale and primal-dual family of interior point methods is investigated. The approaches were applied to a set of real world linear programming problems, with different levels of guidance from the neural networks. The integrated approaches provide promising results, indicating that there may be a place for neural networks in solving large optimization problems.

Esta tese foi financiada pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

A Luisma

AGRADECIMENTOS

A Luisma, meu companheiro e amigo.
A minha filha Mônica, pela alegria de todos os dias.
Aos meus pais e a toda a minha família, pelo carinho e confiança.
Aos meus orientadores Christiano e Aurelio, pela amizade e excelente orientação.
Aos meus amigos e tios cubanos Oda, Oby, Ana, Raúl, Edy, Ileana, Alio e Sahudy, por estarem sempre por perto.
Aos companheiros de laboratório, pela amizade e os cafés.
À Faculdade de Engenharia Elétrica e de Computação, pela grande oportunidade que me deu.
A CAPES e FAPESP, pelo apoio financeiro.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

CAPÍTULO 1.

APRESENTAÇÃO

A utilização de redes neurais para solução de problemas de otimização foi proposta por Hopfield, em trabalho conjunto com Tank [Tank e Hopfield 1986]. Desde então, muitos pesquisadores têm explorado a possibilidade de resolver problemas de otimização com abordagens por redes neurais.

Embora a maioria destes trabalhos seja recente, a análise das contribuições sugere que houve pouca fertilização recíproca entre as áreas de otimização e redes neurais. Todas as abordagens citadas foram aplicadas a exemplos ilustrativos pequenos, que poderiam ser resolvidos sem auxílio de qualquer recurso computacional; estão muito distantes dos problemas reais de grande porte abordados por pesquisadores de áreas de otimização [Adler *et al.* 1989; Gay 1985]. Por outro lado, a comunidade que atua em áreas de otimização bem estabelecidas não tem demonstrado entusiasmo com a idéia de incorporar aspectos de computação através de redes neurais à solução de problemas.

Este trabalho procura identificar interseções mais amplas entre as áreas de redes neurais e métodos de pontos interiores. Redes de Hopfield modificadas [Silva 1997] são utilizadas para obter inicializações avançadas para métodos de pontos interiores afins escala [Tsuchiya 1996] e primais-duais [Wright 1996]. As metodologias propostas são aplicadas a um conjunto de problemas reais de otimização da biblioteca *Netlib* [Gay 1985].

Para a aplicação de redes de Hopfield modificadas na obtenção de inicializações avançadas foram introduzidos aperfeiçoamentos utilizando resultados recentes de álgebra matricial. Estes aperfeiçoamentos permitiram a utilização destas redes na solução de problemas reais de otimização nunca antes abordados na literatura.

As redes de Hopfield modificadas aperfeiçoadas calculam pontos iniciais avançados. Estes pontos são entregues ao método de pontos interiores que continua o processo de otimização. O estudo de caso realizado para cada método, utilizando um conjunto de problemas da biblioteca *Netlib*, mostrou que foi possível diminuir o número de iterações do método de otimização na maioria dos casos. Esta redução da trajetória nos permitiu extrair indicadores sobre as possibilidades das abordagens propostas para a solução de problemas reais de otimização.

Esta dissertação está dividida em sete capítulos e um apêndice.

O Capítulo 2 faz um resumo de conceitos de otimização e redes neurais utilizados neste trabalho. Apresenta também uma revisão bibliográfica das abordagens para solução de problemas de otimização através de redes neurais.

O Capítulo 3 descreve os métodos de pontos interiores para otimização linear utilizados nas abordagens propostas neste trabalho. Duas classes de métodos de pontos interiores são apresentadas: os métodos afins escala e os métodos primais duais. Os sistemas lineares oriundos de métodos de pontos interiores e suas similaridades com os sistemas lineares das redes de Hopfield modificadas são também discutidos. Outros pontos discutidos no capítulo são a inicialização “à quente” em métodos de pontos interiores, pré-processamento e uma breve introdução sobre os problemas do *Netlib*.

No Capítulo 4, são discutidas as redes de Hopfield modificadas para solução de problemas de otimização, propostas por Silva [1997]. O capítulo apresenta também aperfeiçoamentos na metodologia que viabilizaram sua aplicação à solução de problemas reais de otimização.

Alternativas de cooperação entre redes de Hopfield e métodos afins escala de pontos interiores são discutidas no Capítulo 5. O desempenho das alternativas propostas é avaliado em aplicações a problemas reais de otimização da biblioteca *Netlib*.

O Capítulo 6 apresenta propostas de cooperação entre redes de Hopfield e métodos primais-duais de pontos interiores. De forma análoga aos métodos afins escala, o desempenho dos métodos é avaliado em aplicações a um conjunto de problemas do *Netlib.*

Discussões, conclusões e alternativas de desdobramentos são apresentadas no Capítulo 7.

O Apêndice A apresenta uma abordagem em que redes neurais e algoritmos genéticos são conjugados para resolver problemas irrestritos de otimização.

O Apêndice B registra os artigos associados a este trabalho.

CAPÍTULO 2.

OTIMIZAÇÃO LINEAR E REDES NEURAIS

Este capítulo discute tópicos gerais sobre otimização linear e redes neurais artificiais utilizados neste trabalho. Apresenta também uma revisão bibliográfica sobre o uso de redes neurais para a solução de problemas de otimização.

2.1 OTIMIZAÇÃO

A área de otimização estuda o conjunto de conhecimentos matemáticos para a minimização (ou maximização) de uma função f ($f : \Re^n \rightarrow \Re$) em um espaço caracterizado por um conjunto de restrições.

$$\begin{aligned} \min_{x} \quad & f(x) \\ sujeito\ a \quad & g(x) = 0 \\ & h(x) \le 0 \\ & x \in S \end{aligned} \tag{2.1}$$

onde:

$x \in \Re^n$ é o vetor de variáveis do problema;

$g : \Re^n \to \Re^p$;

$h : \Re^n \to \Re^q$;

$S \in \Re^n$.

As funções que definem o problema de otimização podem ser lineares ou não-lineares. As características do problema levam a metodologias de resolução distintas. A otimização linear trata problemas em que todas as funções envolvidas são lineares e S é um politopo em $\Re^n$ ($S = \{x : \underline{x} \le x \le \overline{x}\}$).

2.1.1 Otimização Linear

Um problema de otimização linear, normalmente denominado problema de "programação linear", pode ser apresentado na forma padrão do problema primal:

$$\begin{aligned} \min_{x} \quad & c^T \cdot x \\ sujeito\ a \quad & A \cdot x = b \\ & x \ge 0 \end{aligned} \tag{2.2}$$

onde $c \in \Re^n$ é o vetor de custo do problema, $A \in \Re^{m \times n}$ é a matriz de restrições de m linhas e n colunas, $b \in \Re^m$ é o vetor das restrições e $x \in \Re^n$ é o vetor de variáveis do problema. Para facilitar a apresentação das idéias propostas neste trabalho, consideram-se nulos os limites inferiores das variáveis e não se consideram limites superiores (i.e.

$x \geq 0$), no entanto, todos os conceitos são facilmente generalizados para situações de variáveis com limites inferiores diferente de zero e com limites superiores (i.e. $\underline{x} \leq x \leq \bar{x}$). Um ponto x é denominado factível quando satisfaz todas as restrições do problema 2.2.

Associado ao problema primal (2.2), define-se o problema dual [Luenberger 1984] representado por:

$$\begin{aligned} \max_{y} \quad & b^T \cdot y \\ sujeito \;\; a \quad & A^T \cdot y \leq c \end{aligned} \tag{2.3}$$

onde $y \in \Re^m$.

A Fig. 2.1 representa um politopo no $\Re^2$, definido a partir do conjunto de restrições definido a seguir:

$$\begin{aligned} & 0 \leq x_1 \leq 3 \\ & 0 \leq x_2 \leq 2 \\ & x_1 + x_2 \leq 4 \end{aligned} \tag{2.4}$$

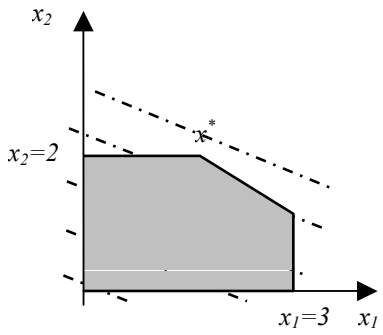

Fig. 2.1. Politopo no $\Re^2$

Uma "solução básica" em um problema de otimização linear corresponde a um vértice do politopo definido pelas restrições [Luenberger 1984]. Os vértices são formados pela interseção de duas ou mais restrições do problema.

2.1.2 Teorema Fundamental da Programação Linear

Teorema Fundamental da Programação Linear: Dado um problema linear na forma padrão, onde A é uma matriz $m \times n$ de posto m, as afirmações a seguir são verdadeiras.

i. Se existe uma solução factível, existe uma solução básica factível;

ii. Se existe uma solução ótima factível, existe uma solução ótima básica factível.

Em outras palavras, o teorema fundamental da programação linear estabelece que é suficiente procurarmos soluções ótimas no subconjunto de soluções formado por soluções básicas. É fácil verificar [Luenberger 1984] que o número de soluções básicas é caracterizado pela Equação (2.5) a seguir.

$$\binom{n}{m} = \frac{n!}{m! \cdot (n-m)!}, \tag{2.5}$$

Embora o número de soluções básicas seja finito, ele aumenta em uma relação fatorial com o tamanho do problema.

2.1.3 Método Simplex

O método simplex é o método mais conhecido para a solução de problemas lineares e foi proposto por Dantzing, em 1947 [Luenberger 1984]. O simplex se movimenta de um vértice a outro do politopo do problema, sempre melhorando o valor da função objetivo; caso não consiga melhorar a função objetivo a solução ótima foi atingida. A Fig. 2.2 ilustra a procura de solução ótima através do método simplex.

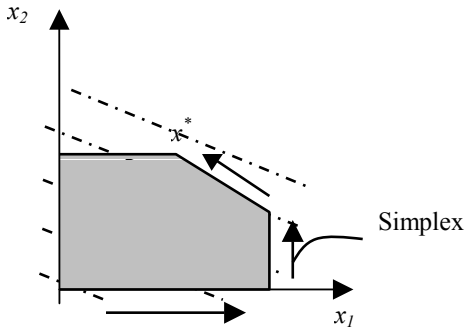

Fig. 2.2. Movimentação do Simplex

Teoricamente, o método simplex pode percorrer todas as soluções básicas do problema no processo de busca de soluções ótimas — existem exemplos que ilustram esta possibilidade [Bazaraa *et al.* 1997]. Considerando-se o número de soluções básicas expresso pela Equação (2.5), observa-se que o esforço de cálculo pode crescer numa relação fatorial com a dimensão do problema.

Deve-se observar que, na maior parte das aplicações práticas, o esforço de cálculo do método simplex é proporcional ao número de restrições do problema. No entanto, o "mau comportamento" na situação de "pior caso", descrita no parágrafo anterior, motivou a pesquisa por métodos alternativos.

O matemático russo Khachiyan [1979] propôs o método dos elipsóides, com melhores propriedades teóricas que o método simplex em análises de pior caso. Foi uma vantagem teórica; implementações práticas do método apresentaram resultados muito inferiores aos obtidos com o simplex. No entanto, a supremacia do simplex viria ser abalada na década seguinte, quando foram propostos os métodos de pontos interiores.

2.1.4 Métodos de Pontos Interiores

Em 1985, Karmarkar formalizou o conceito de métodos de pontos interiores. Nos anos seguintes, novos métodos de pontos interiores foram propostos indicando que os

métodos de pontos interiores apresentavam melhor comportamento do que o simplex, na maior parte dos problemas de grande porte. Em 1989, Adler e co-autores formalizaram o método dual afim escala, onde o problema dual é resolvido a partir de um ponto inicial dual factível. Em seguida, foram propostos os métodos primais-duais, onde os problemas primal e dual são resolvidos simultaneamente, a partir de pontos iniciais primais e duais não necessariamente factíveis. O método primal-dual preditor-corretor [Monteiro *et al.* 1990; Mehrotra 1992] é considerado o método mais eficiente para abordagem de problemas genéricos (i.e., sem estruturas particulares).

Os métodos de pontos interiores se movimentam no interior da região de factibilidade, como ilustra a Fig. 2.3. Este comportamento contrasta com o método simplex que se movimenta entre soluções básicas através da fronteira.

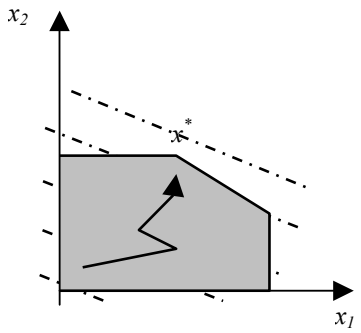

Fig. 2.3. Movimentação dos Métodos de Pontos Interiores.

2.2 REDES NEURAIS

Uma rede neural é definida por Haykin [1999] como um sistema massivamente paralelo e distribuído, formado por unidades de processamento simples chamadas neurônios. As redes neurais adquirem o conhecimento do ambiente através de processos de aprendizado.

O ritmo de pesquisas e entusiasmo com redes neurais aconteceu em três grandes ciclos ao longo dos últimos cinqüenta anos. O primeiro período de entusiasmo ocorreu na década de *40*, desdobrando-se através dos anos *50*. Foi alimentado pelo trabalho de McCulloch e Pitts [1943], onde foi proposto um modelo matemático para o neurônio.

O segundo ciclo foi protagonizado pelo teorema de convergência do *perceptron* [Rosenblatt 1958], com desdobramentos que pareciam fazer crer que as redes neurais, adequadamente concebidas, seriam capazes de lidar com qualquer problema. Esse entusiasmo foi arrefecido com um trabalho de Minsky e Papert [1969], mostrando que redes *perceptron* de uma única camada eram incapazes de resolver alguns problemas simples.

Um novo ciclo de entusiasmo, que se desdobra e cresce até hoje, ganhou ímpeto com o trabalho de Hopfield [1982], que importou da física o conceito de função de energia para explicar o comportamento de redes recorrentes com conexões simétricas. Esta classe de redes neurais passou a ser denominada de redes de Hopfield.

Nas seções que seguem, serão apresentados alguns pontos relacionados com redes neurais que são de interesse para a apresentação deste trabalho.

2.2.1 O Neurônio Artificial

A primeira proposta de modelagem matemática do neurônio foi proposta por McCulloch e Pitts em 1943. O neurônio foi definido como unidade de processamento simples que realiza uma função binária, tudo ou nada (*1* ou *0*). Anos mais tarde Rosenblatt [1958] introduz o *perceptron*, uma nova arquitetura que possuía pesos nas interconexões e era capaz de solucionar problemas linearmente separáveis.

O *perceptron* é caracterizado pelos seguintes elementos:

- Um vetor de entrada;
- Um vetor de pesos *w* que amplifica a entrada;

- O *bias* ou limiar de entrada θ, que representa a intensidade mínima que o neurônio deve receber para produzir uma resposta;
- Uma função de ativação $f(y)$, que determina a forma de resposta do neurônio.

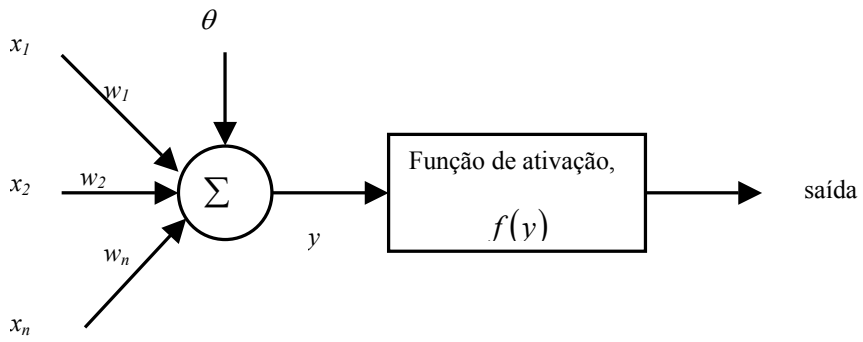

Fig. 2.4. O *Perceptron*

As entradas do neurônio x_i são multiplicadas por seus respectivos pesos w_i, $i = 1,2,\ldots,n$ e somadas, pelo neurônio, ao limiar θ.

$$y = x^T \cdot w + \theta . \tag{2.6}$$

Sobre *y*, é aplicada uma função *f* chamada de função de ativação, na forma:

$$saída = f(y) . \tag{2.7}$$

2.2.2 Função de Ativação

Um neurônio é ativado quando um conjunto de entrada é aplicado; a forma com que a resposta é efetivada é determinada pela função de ativação. Este conceito foi definido por McCulloch e Pitts [1943] na sua proposta inicial do neurônio artificial. Estas funções determinam as saídas dos neurônios e, com isto, os valores das entradas para os próximos neurônios.

Existem vários tipos de função de ativação. Nesta seção, apresentaremos algumas delas: função binária, função linear, função rampa, função logística e tangente hiperbólica.

Função binária. É a função utilizada por McCulloch e Pitts [1943] na definição do neurônio artificial, sua expressão matemática pode ser caracteriza na forma a seguir:

$$f(y)=\begin{cases}1 & se\ \ y\geq 0\\ 0 & se\ \ y<0\end{cases}. \tag{2.8}$$

Função linear. Caracterizada pela Equação (2.9):

$$f(y)=a\cdot y+b \tag{2.9}$$

onde a e b são constantes pré-definidas.

Função rampa. É definida pela expressão (2.10); ilustrada na Fig. 2.5.

$$f(y)=\begin{cases}min & se\ \ y\geq Lim\ \ inf\\ a\cdot y+b & se\ \ Lim\ \ inf<y<Lim\ \ sup\\ max & se\ \ y\geq Lim\ \ sup\end{cases} \tag{2.10}$$

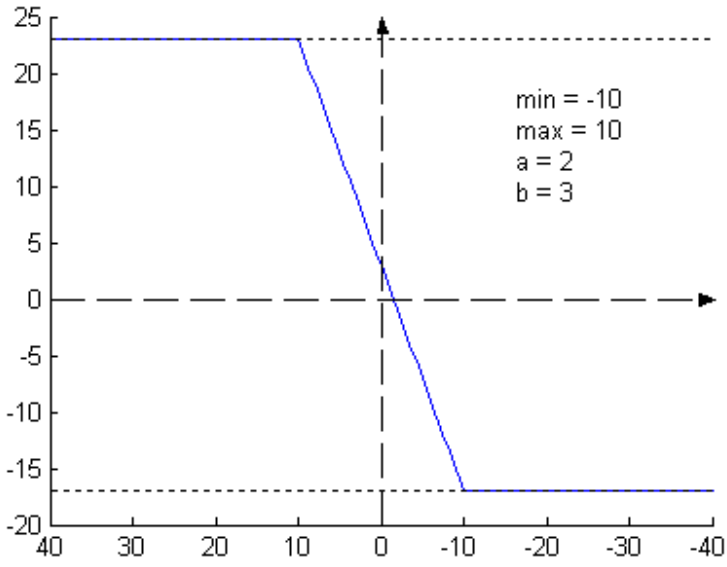

Fig. 2.5. Função Rampa

Função Logística. Definida pela expressão matemática (2.11) e ilustrada na Fig. 2.6.

$$f(y) = \frac{1}{1 + e^{-a \cdot y}} \tag{2.11}$$

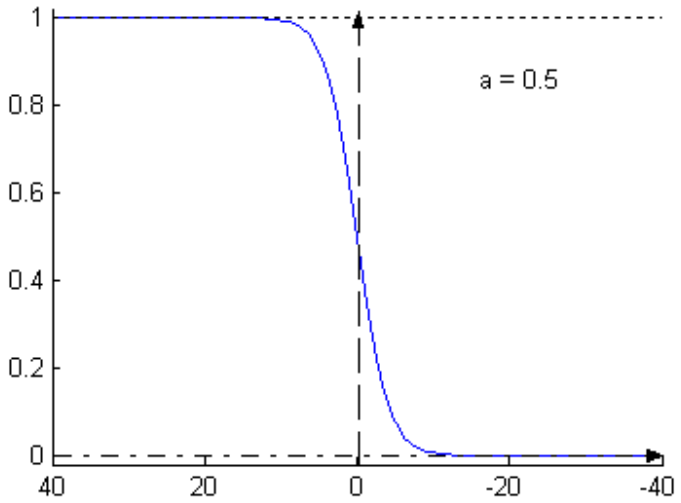

Fig. 2.6. Função Logística

Função Tangente Hiperbólica. Caracterizada pela expressão matemática (2.12) e ilustrada na Fig. 2.7.

$$f(y) = tanh(a \cdot y) = \frac{e^{a \cdot y} - e^{-a \cdot y}}{e^{a \cdot y} + e^{-a \cdot y}} \tag{2.12}$$

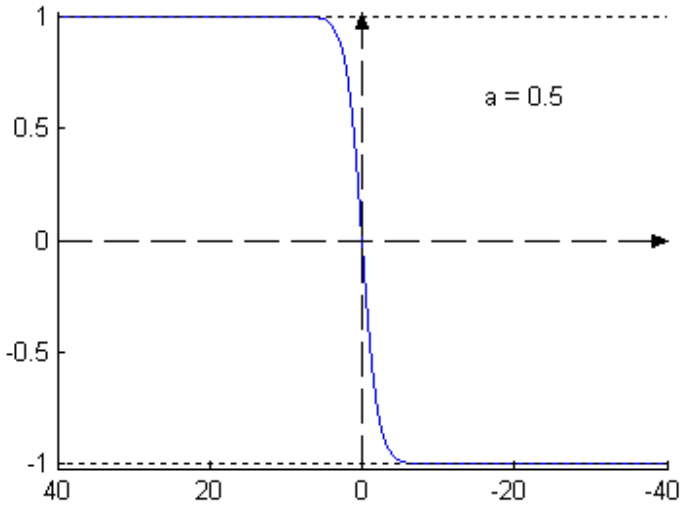

Fig. 2.7. Função Tangente Hiperbólica

2.2.3 Arquitetura da Rede

A arquitetura de uma rede neural é determinada pela forma em que são definidas as conexões entre os neurônios. Basicamente, uma rede é dividida em camadas de neurônios; geralmente, possuem uma camada de entrada, camadas intermediárias e uma camada de saída; em alguns casos, não existem camadas intermediárias. Podemos identificar três classes de arquitetura de redes neurais [Haykin 1999]: redes *feedforward* de uma única camada, redes *feedforward* de múltiplas camadas e redes recorrentes.

Redes *feedforward* de uma única camada. Esta rede possui apenas uma camada de entrada e uma camada de saída. A sua denominação *feedforward*, ou acíclica, decorre do sentido de propagação do sinal, da camada de entrada para a camada de saída (e não vice-versa), não ocorrendo ciclos. Na camada de entrada, não ocorre nenhum processamento, são neurônios lineares (normalmente possuem uma função de ativação linear, $g(x) = x$); todo o processamento ocorre nos neurônios da camada de saída. Esta rede pode ser utilizada para resolver problemas linearmente separáveis [Haykin 1999].

Redes *feedforward* de múltiplas camadas. Diferencia-se da anterior pela presença de camadas intermediárias. A saída de cada camada é dada como entrada da próxima camada. Estas redes podem ser totalmente conectadas quando todos os neurônios da camada anterior são conectados com todos os neurônios da camada seguinte, ou parcialmente conectadas, quando não estão presentes todas as conexões. As camadas intermediárias aumentam a capacidade de processamento da rede [Haykin 1999].

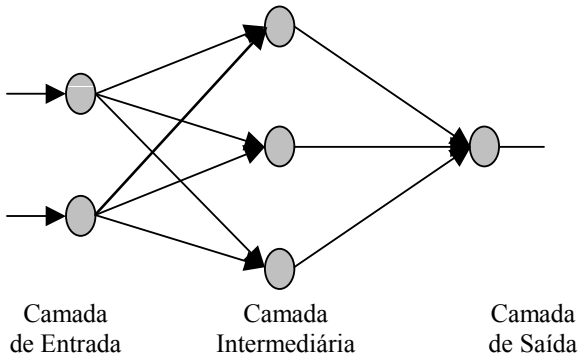

Fig. 2.8. Rede Multicamada Totalmente Conectada

A Fig. 2.8 representa uma rede neural multicamada totalmente conectada, com dois neurônios na camada de entrada, uma camada intermediária de três neurônios e um neurônio na camada de saída.

Redes Recorrentes. As redes recorrentes se diferenciam das redes *feedforward* por existir pelo menos uma conexão de realimentação—isto é, as saídas dos neurônios são re-alimentadas para outros neurônios da rede. Pode-se imaginar que se deseja simular um processo em que o próximo estado depende do estado anterior; as conexões de realimentação permitem essa relação. As redes recorrentes podem ou não ter camadas de neurônios intermediárias.

As redes de Hopfield, ilustradas na Fig. 2.9, são as redes recorrentes mais conhecidas. Estas redes possuem uma única camada, que realiza as funções das camadas de entrada e de saída; não existem camadas intermediárias. Todos os neurônios são realimentados, mas não ocorre auto-realimentação.

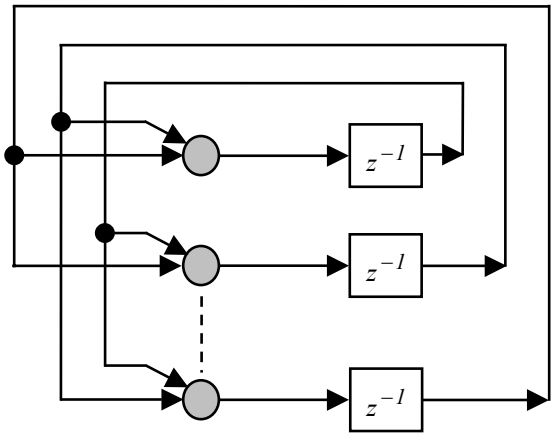

Fig. 2.9. Rede de Hopfield

2.2.4 Otimização nas Redes Multicamadas e Dinâmica das Redes de Hopfield

No processo de treinamento supervisionado em redes neurais multicamadas e na dinâmica das redes de Hopfield podem ser identificados processos de minimização de uma função não-linear.

O processo de treinamento supervisionado em redes neurais multicamadas ocorre através da atualização dos pesos da rede. A partir do estado inicial, definido pelos valores iniciais dos seus parâmetros (pesos da rede e constantes das funções de ativação), a cada iteração é aplicado um padrão de entrada com saída conhecida, propagado através das camadas até a saída da rede. A partir da comparação entre a saída obtida e a saída desejada (saída conhecida do padrão de entrada), é calculado o erro quadrático que mede a distância entre essas saídas. Este erro é retro-propagado e os pesos da rede são atualizados, através de um procedimento de otimização. O processo é repetido, até que o erro quadrático seja minimizado. Quando isto ocorre, a rede neural

conseguiu aprender adaptando-se ao ambiente imposto pelos padrões de entrada. Pode-se interpretar que o processo de aprendizado supervisionado em uma rede neural multicamada ocorre através da minimização de uma função não-linear nos parâmetros da rede (função de erro quadrático) através do método de otimização adotado.

O método adotado para minimização de erro quadrático no processo de treinamento é escolhido em função dos problemas abordados e dos padrões de treinamento escolhidos. A título de exemplificação, pode-se destacar alguns métodos adotados: gradiente [Haykin 1999], gradiente conjugado [Haykin 1999], Newton [Haykin 1999], métodos de pontos interiores [Szymanski *et al.* 1998; Trafalis e Couellan 1994; Lemmon e Szymanski 1994] e algoritmos genéticos [Ku *et al.* 1995; Iyoda *et al.* 1999; Velazco e Lyra 2002].

Por outro lado, a dinâmica discreta das redes de Hopfield [Haykin 1999] é descrita por um sistema não-linear de equações diferenciais ordinárias e por sua função de energia, caracterizada como uma função de Lyapunov. A forma com que a função de energia (uma função de Lyapunov) é desenvolvida garante a estabilidade do sistema para qualquer estado inicial; em outras palavras, a função é monotonicamente decrescente para um estado de equilíbrio da rede [Hertz *et al.* 1991]. Dado um estado inicial, a trajetória de estados sucessivos é obtida por um processo de relaxação da dinâmica não-linear que converge a um atrator levando o sistema a um estado estacionário. Existem quatro possíveis atratores: pontos fixos (mínimos ou pontos de equilíbrios), soluções periódicas , soluções quase-periódicas e caos.

O tema central deste trabalho é a solução de problemas de otimização através de redes neurais. Neste caso, o problema a ser abordado por redes neurais é um problema de otimização como discutido na seção 2.1. Um ponto importante para compreensão dos temas desenvolvidos é a separação entre o problema abordado e os aspectos de otimização para atualização de pesos em redes multicamadas ou na dinâmica de redes de Hopfield; no caso, o problema abordado no trabalho é um “problema de otimização”,

mas que poderia ser, por exemplo, um problema de identificação de sistemas dinâmicos ou controle de processos.

2.3 Revisão da Bibliografia sobre Solução de Problemas de Otimização por Redes Neurais

O primeiro trabalho para a resolução de problemas de otimização foi proposto por Tank e Hopfield em 1986. Nesta rede, definida para problemas lineares com restrições, é construída uma função de energia a partir da função objetivo do problema de otimização e das restrições que são incorporadas como termos de penalidade.

Um aspecto inconveniente com as redes desenvolvidas por Tank e Hopfield é o fato de não haver demonstração formal de que o ponto de equilíbrio corresponde a uma solução ótima do problema original, satisfazendo as condições de otimalidade de Kuhn-Tucker [Maa e Shanblatt 1992a].

Dois anos após a publicação do trabalho de Tank e Hopfield, Kennedy [1988] fez uma análise das redes neurais recorrentes para a resolução de problemas de otimização não-linear. Nesta proposta, estende o modelo de Tank e Hopfield para a resolução de problemas de otimização não-linear. O problema com restrições é transformado em um problema irrestrito através de funções de penalidade, e é criada uma função de Lyapunov. Este tipo de transformação possui o inconveniente de que, quando a solução do problema está sobre a fronteira da região de factibilidade definida pelas restrições, a rede converge para uma solução aproximada, fora da região de factibilidade.

Em 1990, Rodriguez e co-autores propuseram uma abordagem similar à de Tank e Hopfield [1986] e Kennedy [1988], para resolver problemas de otimização restritos e irrestritos. Os problemas com restrições são transformados em problemas irrestritos a partir de funções de penalidade, e resolvidos pelo método do gradiente de maior descida [Luenberger 1984].

Em 1992, Maa e Shanblatt [1992b] propuseram uma abordagem alternativa para a solução de problemas de infactibilidade apresentado nas redes de Hopfield quando a solução ótima encontra-se na fronteira da região de factibilidade. Este método, chamado de otimização em duas fases, resolve problemas não-lineares, com e sem restrições, usando o gradiente com passo fixo como método de minimização da função de energia.

Para a resolução de problemas de programação dinâmica, Chiu e co-autores [1991] utilizaram também redes recorrentes. A partir das restrições impostas por este tipo de problema, obtém-se uma função de energia que garante a estabilidade da rede e a convergência para um ponto de equilíbrio. O trabalho faz uma análise dos parâmetros da função de energia para melhorar o comportamento da rede e definir condições iniciais mais atraentes. O conjunto desses procedimentos permite acelerar a convergência.

Zhang e co-autores [1992] propuseram um método de segunda ordem que utiliza as redes recorrentes de Hopfield para solução de problemas de otimização linear com restrições. Essa abordagem utiliza os multiplicadores de Lagrange para a obtenção da função de energia e garante que as soluções pertencem ao "subespaço válido" (conceito também usado por Silva [1997]). A função obtida é uma "função de Lyapunov", fazendo com que a rede seja estável. O método de Newton é utilizado para minimizar a função de energia.

Osowski [1992] utiliza uma rede neural recorrente para resolver problemas de otimização não-linear com restrições lineares de igualdade, mas não acrescenta aspectos novos significativos em relação às abordagens anteriores. O problema é transformado em um problema sem restrições e no processo de otimização da função de energia são utilizados os métodos do gradiente e de Newton.

Xia e Wang [1995; 1996] apresentaram uma rede neural para solução de problemas de otimização linear. A rede neural é definida a partir das condições de otimalidade [Luenberger 1984] do problema linear. Em 1996 [Xia 1996a], esta rede é estendida para a solução de problemas de otimização quadrática com restrições lineares. Em 1998, a abordagem é novamente estendida, para solução de problemas de otimização com e sem

restrições [Xia 1998]. Esta rede possui convergência global para a solução ótima [Xia 2000].

Outras publicações com metodologias semelhantes utilizam redes de Hopfield para a resolução de problemas de otimização linear [Lillo *et al.* 1993], não-linear [Chua 1990] e quadrática [Bouzerdoum e Pattison 1993; Pérez-Ilzarbe 1998; Wu e Tam 1999].

Na aplicação de redes neurais multicamadas para otimização, podem ser citadas as abordagens de Romero [1993; 1996] e de Reifman e co-autores [1999]. Ambas as abordagens transformam o problema de otimização com restrições em um problema irrestrito, onde as restrições são adicionadas como termos de penalidade, criando uma nova função objetivo irrestrita. As variáveis são identificadas com os neurônios e a função de erro quadrático é substituída pela nova função objetivo. A otimização é realizada sobre os pesos da rede. A abordagem de Romero foi modificada para a definição da abordagem Neuro-Evolutiva [Velazco e Lyra 2002] descrita no Apêndice A, na qual algoritmos genéticos realizam a atualização dos pesos. Esta nova abordagem mostrou melhores resultados nos problemas tratados, em comparação à abordagem de Romero.

Barbosa e Carvalho [1990] fizeram uma exploração preliminar da possibilidade de gerar, através de uma rede neural multicamada, uma seqüência de pontos interiores que convergem para a solução de um problema de otimização linear.

Outro tipo de redes a serem citadas na resolução de problemas de otimização são as redes auto-organizáveis de Kohonen. Pham e Karaboga [2000] apresentam uma abordagem para a solução de problemas de locação de módulos em circuitos.

Silva [1997] utiliza uma rede de Hopfield para solucionar problemas de otimização linear, não-linear [Silva *et al.* 1998], quadrática e dinâmica [Silva *et al.* 1999]. A rede possui um controlador nebuloso [Pedrycz e Gomide 1998] para o cálculo do passo adotado no método de minimização da função de energia.

Embora a maioria destes trabalhos seja recente, a análise das contribuições sugere que houve pouca fertilização recíproca entre as áreas de otimização e redes neurais. Todas as abordagens citadas foram aplicadas a exemplos ilustrativos pequenos, que poderiam ser resolvidos sem auxílio de qualquer recurso computacional; estão muito distantes dos problemas reais de grande porte abordados por pesquisadores de áreas de otimização [Adler *et al.* 1989; Gay 1985].

Velazco e co-autores [2002a; 2002b; 2003] desenvolveram abordagens em que redes de Hopfield modificadas [Silva 1997] são utilizadas para encontrar pontos iniciais de boa qualidade para os métodos de pontos interiores. Vale ressaltar que eventuais problemas de convergência da abordagem de Hopfield inexistem nas abordagens desenvolvidas por Velazco e co-autores. A rede desenvolvida nesses trabalhos não é utilizada na solução do problema de otimização, mas apenas para a obtenção de pontos inicias que são entregues ao método de pontos interiores, que finalizam a solução do problema. Esses aspectos são discutidos nos próximos capítulos.

CAPÍTULO 3.

MÉTODOS DE PONTOS INTERIORES

Duas classes de métodos de pontos interiores são apresentadas neste capítulo: os métodos afins escala e os métodos primais-duais. Os métodos afins apresentados são o método primal afim escala [Dikin 1967; Tsuchiya 1996] e o método dual afim escala [Adler *et al.* 1989]. Na categoria dos métodos primais-duais discute-se o método primal-dual clássico [Wright 1996] e o método preditor-corretor [Monteiro *et al.* 1990; Mehrotra 1992].

O capítulo discute também outros aspectos importantes relacionados com métodos de pontos interiores: obtenção de pontos para inicialização do processo de solução, solução de sistemas lineares oriundos desses métodos, esparsidade das matrizes e pré-processamento. Ao final do capítulo, faz-se uma breve apresentação do conjunto de problemas agrupados na biblioteca *Netlib*, utilizado nos estudos de casos discutidos no trabalho.

3.1 CONSIDERAÇÕES GERAIS

Um problema de otimização linear pode ser apresentado na forma padrão do problema primal, como segue:

$$\begin{aligned} &min && c^T \cdot x \\ &sujeito\ a && A \cdot x = b \\ & && x \geq 0 \end{aligned} \tag{3.1}$$

onde $A \in \Re^{m \times n}$ é a matriz de restrições de *m* linhas e *n* colunas, $c \in \Re^n$ é o vetor de custo do problema, $b \in \Re^m$ é o vetor das restrições e $x \in \Re^n$ é o vetor de variáveis do problema, restritas a valores não negativos ($x \geq 0;\ x_i \geq 0, i = 1,2,\ldots,n$). Associado ao problema primal (3.1), define-se o problema dual representado por:

$$\begin{aligned} &max && b^T \cdot y \\ &sujeito\ a && A^T \cdot y \leq c \end{aligned} \tag{3.2}$$

onde $y \in \Re^m$ é o vetor de variáveis. Eliminando-se as restrições de desigualdade com a adição do vetor de variáveis de folga *z*, obtém-se o problema a seguir:

$$\begin{aligned} &max && b^T \cdot y \\ &sujeito\ a && A^T \cdot y + z = c \\ & && z \geq 0 \end{aligned} \tag{3.3}$$

onde $z \in \Re^n$ está restrito a valores não-negativos.

O **Teorema Fundamental da Dualidade** em programação linear estabelece condições de otimalidade para o par de problemas primal e dual [Luenberger 1984]:

Um ponto $\left(x^*, y^*, z^*\right)$ é o ótimo simultaneamente do primal e do dual se satisfaz as condições a seguir.

1. Factibilidade do primal: $b - A \cdot x^* = 0, \quad x^* \geq 0$.
2. Factibilidade do dual: $c - A^T \cdot y^* - z^* = 0$.

3. Complementaridade: $x_i^* \cdot z_i^* = 0, \quad \forall i = 1,2,\ldots,n$.

Os métodos de pontos interiores, como sugere a denominação, procuram uma solução ótima do problema através de buscas no interior do espaço de factibilidade. Formalmente, diz-se que um ponto é interior quando está dentro do politopo formado pelas restrições de desigualdade do problema. Para o problema primal (definido em (3.1)) esta propriedade é caracterizada por $x > 0$. Para o problema dual (3.3), os pontos interiores satisfazem a condição $z > 0$.

3.2 MÉTODOS AFINS ESCALA

Nesta seção, são discutidos dois métodos afins: o método primal afim escala e o método dual afim escala.

3.2.1 Método Primal Afim Escala

A motivação do método é calcular o próximo ponto primal a partir de uma estimativa das variáveis duais. Esta estimativa é obtida a partir da solução do problema definido a seguir:

$$\begin{aligned} \min_{(y,z)} \quad & f(y,z) = \frac{1}{2} \cdot \|X \cdot z\|^2 \\ \textit{sujeito a} \quad & z = c - A^T \cdot y \end{aligned} \tag{3.4}$$

sendo X uma matriz diagonal $n \times n$ onde os elementos da diagonal são os valores do vetor x.

É fácil verificar que o problema (3.4) procura encontrar o vetor (y,z) factível que se aproxime, ao máximo, da condição de complementaridade estabelecida pelo Teorema Fundamental da Dualidade (i.e., $x_i^* \cdot z_i^* = 0, \quad \forall i = 1,2,\ldots,n$).

Para obter a estimativa das variáveis duais (y,z) a partir da solução de (3.4), pode-se substituir z na função objetivo, obtendo-se o problema irrestrito (3.5).

$$\min_{y} f(y) = \frac{1}{2} \cdot \left\| X \cdot \left(c - A^T \cdot y \right) \right\|^2 \tag{3.5}$$

Como (3.5) é quadrático, pode-se calcular y a partir das condições necessárias de otimalidade. Fazendo $\nabla f(y) = 0$, vem:

$$\nabla f(y) = -(A \cdot X) \cdot \left(X \cdot c - X \cdot A^T y \right) = 0 . \tag{3.6}$$

Logo,

$$y = \left(A \cdot X^2 \cdot A^T \right)^{-1} \cdot A \cdot X^2 \cdot c . \tag{3.7}$$

Usando a definição do problema dual (3.3), tem-se:

$$z = c - A^T \cdot y . \tag{3.8}$$

Com esse valor estimado das variáveis duais é calculada a direção de descida factível $\Delta x = -(X)^2 \cdot z$ [Dikin 1967]. Usando-se esta direção, encontra-se o novo ponto x.

É fácil verificar que a direção $\Delta x = -(X)^2 \cdot z$, definida por Dikin [1967], é factível e de descida.

i. Para verificar que a direção é factível, considere um novo ponto $\tilde{x} = x + \alpha \cdot \Delta x$ obtido a partir da direção.

$$\begin{aligned} A \cdot \tilde{x} = A \cdot (x + \alpha \cdot \Delta x) &= A \cdot x - \alpha \cdot A \cdot X^2 \cdot z \\ &= b - \alpha \cdot A \cdot X^2 \cdot \left(c - A^T \cdot y \right) \\ &= b - \alpha \cdot A \cdot X^2 \cdot c + \alpha \cdot A \cdot X^2 \cdot A^T \cdot y \end{aligned}$$

Como $y = \left(A \cdot X^2 \cdot A^T \right)^{-1} \cdot A \cdot X^2 \cdot c$ (Equação 3.7),

$$A \cdot \tilde{x} = b - \alpha \cdot A \cdot X^2 \cdot c + \alpha \cdot A \cdot X^2 \cdot c = b .$$

ii. Para verificar que a direção é de descida, é suficiente mostrar que $c^T \cdot \tilde{x} < c^T \cdot x$.

$$c^T \cdot \tilde{x} = c^T \cdot (x + \alpha \cdot \Delta x) = c^T \cdot x + \alpha \cdot c^T \cdot \Delta x$$

Como $\alpha > 0$, deve-se mostrar que $c^T \cdot \Delta x < 0$. Para isso é suficiente verificar que $c^T \cdot \Delta x = -\left\| X^{-1} \cdot \Delta x \right\|^2 < 0$. De fato, tem-se:

$$\begin{aligned}
\left\| X^{-1} \cdot \Delta x \right\|^2 &= \left(X^{-1} \cdot \Delta x \right)^T \cdot \left(X^{-1} \cdot \Delta x \right) \\
&= \Delta x^T \cdot X^{-2} \cdot \Delta x \\
&= z^T \cdot X^2 \cdot X^{-2} \cdot X^2 \cdot z \\
&= z^T \cdot X^2 \cdot z \\
&= \left(c - A^T \cdot y \right)^T \cdot X^2 \cdot z \\
&= c^T \cdot X^2 \cdot z - y^T \cdot A \cdot X^2 \cdot z \\
&= -c^T \cdot \Delta x - y^T \cdot A \cdot X^2 \cdot \left(c - A^T \cdot y \right) \\
&= -c^T \cdot \Delta x - y^T \cdot \left(A \cdot X^2 \cdot c - A \cdot X^2 \cdot A^T \cdot y \right) \\
&= -c^T \cdot \Delta x
\end{aligned}$$

Logo $c^T \cdot \Delta x = -\left\| X^{-1} \cdot \Delta x \right\|^2$.

Considerando-se os resultados anteriores, o método primal afim escala pode ser resumido nos passos a seguir [Tsuchiya 1996].

A partir de um ponto inicial primal interior factível $x^0 > 0$ e $\tau \in (0,1)$, para $k = 0,1,2,...,$ fazer, até convergir:

1. Calcular a estimativa das variáveis duais, $y^k = \left(A \cdot \left(X^k \right)^2 \cdot A^T \right)^{-1} \cdot A \cdot X^2 \cdot c$ e $z^k = c - A^T \cdot y^k$;
2. Calcular a direção de busca, $\Delta x^k = -\left(X^k \right)^2 \cdot z^k$;
3. Calcular um tamanho de passo apropriado para manter o ponto interior, $\alpha^k = \tau \cdot \min_{\Delta x_i^k < 0} \left\{ -\frac{x_i^k}{\Delta x_i^k} \right\}$;
4. Calcular o novo ponto interior e factível, $x^{k+1} = x^k + \alpha^k \cdot \Delta x^k$.

O passo α^k na direção Δx^k garante que o novo ponto é interior. Observando-se que o cálculo do passo α^k está dividido em duas componentes:

- $\min\limits_{\Delta x_i^k < 0} \left\{ -\dfrac{x_i^k}{\Delta x_i^k} \right\}$, expressão que define o passo máximo na direção Δx^k, de forma a não violar a restrição de não-negatividade;
- τ, parâmetro no intervalo aberto $(0,1)$, usado para garantir que o novo ponto é interior—as implementações normalmente adotam $\tau = 0.9$ ou $\tau = 0.99$.

Um aspecto que pode causar dificuldades de convergência no método primal afim escala é o acúmulo de erros de arredondamento no cálculo sucessivo de x^{k+1} a partir da direção Δx^k; pode ocorrer que a restrição matricial $A \cdot x^{k+1} = b$ deixe de ser estritamente verdadeira.

O método normalmente utiliza os seguintes critérios de convergência, baseados nas condições de otimalidade:

- Factibilidade Primal, $\dfrac{\|b - A \cdot x\|}{1 + \|b\|} \leq \varepsilon$;
- Factibilidade Dual, $\dfrac{\|c - A^T \cdot y - z\|}{1 + \|c\|} \leq \varepsilon$;
- GAP Relativo, $\dfrac{|x^T \cdot z|}{1 + |b^T \cdot y + c^T \cdot x|} \leq \varepsilon$. Esta condição é definida a partir de "GAP" (absoluto),

$$GAP = x^T \cdot z, \tag{3.9}$$

que funciona como medida da distância da condição de complementaridade (condição de otimalidade) para soluções factíveis.

3.2.2 Método Dual Afim Escala

O Método Dual Afim Escala foi o primeiro método de pontos interiores que obteve bons resultados na prática [Adler *et al.* 1989]. Seu princípio de funcionamento é resolver o problema dual a partir de uma estimativa das variáveis primais, partindo de um ponto dual inicial interior e factível. A estimativa é obtida pela definição do seguinte problema, baseado nas condições de otimalidade:

$$\begin{array}{ll} \underset{(x)}{min} & \frac{1}{2}\cdot\|Z\cdot x\|^2 \\ sujeito\ a & A\cdot x = b \end{array} \tag{3.10}$$

sendo Z uma matriz diagonal $n\times n$, onde os elementos da diagonal são os valores do vetor z.

Para o cálculo do mínimo do problema (3.10), define-se o Lagrangeano deste problema [Luenberger 1984]:

$$\underset{(x,w)}{min} \qquad g(x)=\frac{1}{2}\cdot\|Z\cdot x\|^2 + w^T\cdot(b-A\cdot x). \tag{3.11}$$

Derivando-se a função $g(x)$, obtém-se condições necessárias de otimalidade para o problema (3.11):

$$Z^2\cdot x - A^T\cdot w = 0 \tag{3.12}$$

$$b - A\cdot x = 0 \tag{3.13}$$

Utilizando-se as condições de otimalidade (3.12) e (3.13), é possível obter uma estimativa para a variável x:

$$x = Z^{-2}\cdot A^T\cdot\left(A\cdot Z^{-2}\cdot A^T\right)^{-1}\cdot b \tag{3.14}$$

A partir da estimativa da variável x e utilizando-se a direção $\Delta z = -Z^2\cdot x$, tem-se que:

$$\Delta z = -Z^2 \cdot Z^{-2} \cdot A^T \cdot \left(A \cdot Z^{-2} \cdot A^T\right)^{-1} \cdot b \tag{3.15}$$

$$\Delta z = -A^T \cdot \left(A \cdot Z^{-2} \cdot A^T\right)^{-1} \cdot b\,. \tag{3.16}$$

Por outro lado, $\Delta z = -A^T \cdot \Delta y$ para qualquer direção dual factível $(\Delta y, \Delta z)$. Então,

$$\Delta y = \left(A \cdot Z^{-2} \cdot A^T\right)^{-1} \cdot b\,. \tag{3.17}$$

O método pode ser resumido nos seguintes passos [Adler *et al.* 1989]:

A partir de um ponto dual inicial interior factível $\left(y^0, z^0 > 0\right)$ e $\tau \in (0,1)$, para $k = 0,1,2,...,$ fazer até convergir:

1. Calcular a direção de busca, $\Delta y^k = \left(A \cdot \left(Z^k\right)^{-2} \cdot A^T\right)^{-1} \cdot b$, $\Delta z^k = -A^T \cdot \Delta y^k$;
2. Calcular a estimativa das variáveis primais, $x^k = -\left(Z^k\right)^{-2} \cdot \Delta z^k$;
3. Calcular o passo apropriado para manter o ponto interior, $\alpha^k = \tau \cdot \underset{\Delta z_i^k < 0}{min}\left\{-\frac{z_i^k}{\Delta z_i^k}\right\}$ (observa-se que a variável y é livre, não influenciando no cálculo do passo)
4. Calcular o novo ponto interior, $\left(y^{k+1}, z^{k+1}\right) = \left(y^k, z^k\right) + \alpha^k \cdot \left(\Delta y^k, \Delta z^k\right)$.

O cálculo de z pela relação $z^{k+1} = z^k + \alpha^k \cdot \Delta z^k$ acumula erros de arredondamento. Portanto, na prática, obtém-se implementações mais robustas quando z^{k+1} é calculado pela relação $z^{k+1} = c - A^T \cdot y^{k+1}$.

Vale também observar que o cálculo de x^k a cada iteração não é necessário, uma vez que a solução dual pode ser calculada sem esta informação.

3.3 MÉTODOS PRIMAIS-DUAIS

Os métodos primais-duais [Wright 1996] resolvem o problema primal e o problema dual simultaneamente a partir de um ponto inicial primal-dual, não necessariamente factível, mas estritamente positivo (ponto interior). O método é obtido a partir da aplicação do método de Newton ao sistema não-linear $F(x,y,z)$ (Equação 3.18) formado pelas condições de otimalidade, mas desconsiderando-se as restrições de não-negatividade.

$$F(x,y,z)=\begin{pmatrix} A\cdot x-b \\ A^T\cdot y+z-c \\ x^T\cdot z \end{pmatrix}=0. \tag{3.18}$$

Neste trabalho, apresentaremos dois métodos primais-duais: o método primal-dual clássico e o método preditor-corretor. Na definição dos mesmos, é introduzida uma perturbação ou medida de dualidade μ na condição de complementaridade, que permite obter pontos mais próximos do centro do politopo. Com isto, em princípio, pode-se obter uma melhor direção no sentido da otimalidade.

$$x_i\cdot z_i=\mu \tag{3.19}$$

A Fig. 3.1 mostra duas trajetórias do método primal-dual em um problema de *2* variáveis e *2* restrições de desigualdade. A trajetória *A* foi calculada com utilização da perturbação μ (i.e. $\mu>0$), o que corresponde ao método primal-dual clássico. A trajetória *B* foi obtida sem a perturbação (i.e. $\mu=0$), que corresponde ao método primal-dual afim escala [Monteiro *et al.* 1990]. Como se observa na Fig. 3.1, quando a perturbação não é utilizada, as direções obtidas têm a tendência de tangenciar a fronteira; quando a perturbação é utilizada, os pontos obtidos a cada iteração permanecem mais próximos do centro do politopo.

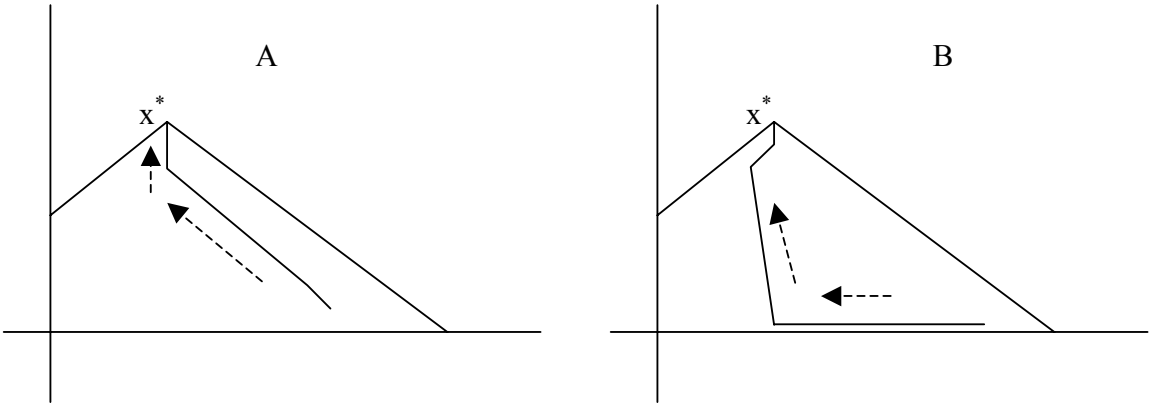

Fig. 3.1. Trajetória do método primal dual: A) Método clássico $\mu > 0$, B) Método afim $\mu = 0$

Os métodos primais-duais não exigem a condição de factibilidade das soluções, enquanto a otimalidade não for alcançada. Portanto, substitui-se o lado direito da Equação (3.18) por um vetor de "resíduos", que caracteriza as distâncias da factibilidade, para os dois primeiros conjuntos de equações, e da condição de complementaridade, para o último conjunto de equações.

$$-F(x,y,z) = \begin{pmatrix} b - A \cdot x \\ c - A^T \cdot y - z \\ x \cdot z \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_c \end{pmatrix} = r \,. \tag{3.20}$$

A partir do ponto $\left(x^k, y^k, z^k\right)$, o próximo ponto é calculado pelo método de Newton, como segue:

$$\left(x^{k+1}, y^{k+1}, z^{k+1}\right) = \left(x^k, y^k, z^k\right) - \alpha \cdot \left(\Delta x^k, \Delta y^k, \Delta z^k\right), \tag{3.21}$$

$$\left(\Delta x^k, \Delta y^k, \Delta z^k\right) = -J^{-1}\left(x^k, y^k, z^k\right) \cdot r^k \,, \tag{3.22}$$

onde $J^{-1}\left(x^k, y^k, z^k\right)$ é o jacobiano do sistema não linear (3.18) e α é o tamanho do passo do método de Newton que possui valor máximo $\alpha = 1$.

3.3.1 Método Primal-Dual Clássico

O método primal-dual clássico ou "seguidor de caminho" [1] [Kojima *et al.* 1989] calcula a direção $(\Delta x, \Delta y, \Delta z)$ através da solução do sistema não-linear a seguir:

$$J(x,y,z)\cdot\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = -F(x,y,z), \tag{3.23}$$

onde $J(x,y,z)$ é definido como:

$$J(x,y,z) = \begin{bmatrix} A & 0 & 0\\ 0 & A^T & I\\ Z^k & 0 & X^k\end{bmatrix}. \tag{3.24}$$

Este método utiliza a perturbação μ na condição de complementaridade, $x_i \cdot z_i = \mu$. Este parâmetro é definido na maioria das implementações como:

$$\mu^k = \sigma^k \cdot \left(\frac{\gamma^k}{n}\right), \tag{3.25}$$

onde σ é fixo—geralmente são utilizados os valores $\sigma = 1/n$ ou $\sigma = 1/\sqrt{n}$. O valor de γ^k é calculado como:

$$\gamma^k = x^{k^T} \cdot z^k. \tag{3.26}$$

Quando introduzimos o novo parâmetro μ^k, o sistema não-linear formado pelas condições de otimalidade assume a forma:

---

[1] Do inglês "path following"

$$-F\left(x^k, y^k, z^k\right) = \begin{bmatrix} b - A \cdot x^k \\ c - A^T \cdot y^k - z^k \\ \mu^k \cdot e - X^k \cdot Z^k \cdot e \end{bmatrix} = \begin{bmatrix} r_p^k \\ r_d^k \\ r_c^k \end{bmatrix}. \tag{3.27}$$

onde *e* é um vetor coluna de valor unitário (da dimensão apropriada).

O sistema para calcular a direção em cada iteração é obtido pelas Equações (3.23), (3.24) e (3.27);

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{bmatrix} \cdot \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{bmatrix} = \begin{bmatrix} rp^k \\ rd^k \\ rc^k \end{bmatrix} = \begin{bmatrix} b - A \cdot x^k \\ c - A^T \cdot y^k - z^k \\ \mu^k \cdot e - X^k \cdot Z^k \cdot e \end{bmatrix}. \tag{3.28}$$

O funcionamento do método primal-dual clássico pode ser resumido na seqüência de passos descrita a seguir.

A partir de um ponto inicial primal-dual interior ($\left(x^0, z^0\right) > 0$ e $y^0 \in \Re^n$), $\sigma \in [0,1)$ e $\tau \in (0,1)$, para $k = 0,1,2,...,$ fazer até convergir:

1. Calcular $\mu^k = \sigma \cdot \left(\frac{\gamma^k}{n}\right)$, onde *n* é a dimensão do vetor *x* e $\gamma^k = \left(x^k\right)^T \cdot z^k$.
2. Calcular a direção de Newton, $\left(\Delta x^k, \Delta y^k, \Delta z^k\right)$.
3. Calcular o tamanho dos passos primal e dual para manter o ponto interior, $\alpha_p^k = min\left(1, \tau \cdot \rho_p^k\right)$, $\alpha_p^k = min\left(1, \tau \cdot \rho_d^k\right)$, $\rho_p^k = \frac{-1}{\min\limits_j \left(\frac{\Delta x_i^k}{x_j^k}\right)}$ e $\rho_d^k = \frac{-1}{\min\limits_j \left(\frac{\Delta z_i^k}{z_j^k}\right)}$.
4. Calcular o novo ponto, $\left(x^{k+1}, y^{k+1}, z^{k+1}\right) = \left(x^k, y^k, z^k\right) + \alpha^k \cdot \left(\Delta x^k, \Delta y^k, \Delta z^k\right)$.

Nas primeiras iterações, μ^k deve ter um valor grande, pois estamos interessados em pontos longes da fronteira. Mas, a cada iteração, à medida que nos aproximamos do ótimo, μ^k vai diminuindo. Esta relação é garantida por γ^k, no cálculo de μ^k pela Equação (3.25). A variável γ^k ou GAP é calculada pela expressão (3.26); vai

diminuindo à medida que nos aproximamos do ótimo (i. e. $\gamma^k \to 0 / k \to \infty$). Pela relação direta que existe entre μ^k e γ^k, então $\mu^k \to 0$ quando $k \to \infty$.

3.3.2 Método Preditor-Corretor

Como já mencionado, o Método Preditor–Corretor [Monteiro *et al.* 1990; Mehrotra 1992] é considerado a abordagem mais eficiente para solução de problemas genéricos de programação linear (em termos de número de iterações e tempos de processamento). Este método, de forma análoga ao método primal-dual clássico, inicializa o processo de otimização com um ponto interior não-factível.

Este método utiliza três componentes para calcular a direção:

1. Uma direção preditora $\left(\Delta\tilde{x}^k, \Delta\tilde{y}^k, \Delta\tilde{z}^k\right)$ ou direção afim, sem a perturbação μ, calculada como segue:

$$\left(\Delta\tilde{x}^k, \Delta\tilde{y}^k, \Delta\tilde{z}^k\right) = -J^{-1}\left(x^k, y^k, z^k\right) \cdot r \tag{3.29}$$

$$\begin{bmatrix} b - A \cdot x^k \\ c - A^T \cdot y^k - z^k \\ Z^k \cdot X^k \cdot e \end{bmatrix} = \begin{bmatrix} r_p^k \\ r_d^k \\ r_a^k \end{bmatrix} = r\,. \tag{3.30}$$

onde $J^{-1}\left(x^k, y^k, z^k\right)$ é o jacobiano do sistema não linear (3.18).

A partir da direção preditora, é calculado um ponto auxiliar $\left(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k\right)$,

$$\begin{aligned} \tilde{x}^k &= x^k + \tilde{\alpha}_p \cdot \Delta\tilde{x} \\ \tilde{y}^k &= y^k + \tilde{\alpha}_d \cdot \Delta\tilde{y} \\ \tilde{z}^k &= z^k + \tilde{\alpha}_d \cdot \Delta\tilde{z}\,. \end{aligned} \tag{3.31}$$

onde $\tilde{\alpha}_p$ e $\tilde{\alpha}_d$ são os passos calculados para a direção preditora, que garantem a interioridade do ponto auxiliar.

2. De acordo com o progresso da direção preditora, é calculada a perturbação μ^k. Se a direção for boa, a perturbação é pequena; caso contrário, é aumentado o peso da direção de centragem (Equação 3.33). Quando $\gamma^k < 1$ (Equação 3.33), existem razões teóricas para colocar uma perturbação da ordem de $\left(\gamma^k\right)^2$ [Tapia e Zhang 1992]. A perturbação é calculada na forma descrita a seguir:

$$\mu^k = \sigma^k \cdot \left(\gamma^k \Big/ n \right), \tag{3.32}$$

onde,

$$\sigma^k = \begin{cases} \left(\dfrac{\tilde{\gamma}^k}{\gamma^k} \right)^3 & se \quad \gamma^k > 1 \\ \dfrac{\gamma^k}{\sqrt{n}} & c.c. \end{cases}, \tag{3.33}$$

$$\tilde{\gamma}^k = \tilde{x}^k \cdot \tilde{z}^k . \tag{3.34}$$

3. Uma direção de correção não-linear é calculada, utilizando-se o sistema análogo ao da expressão (3.27), descrito a seguir [Mehrotra 1992],

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{bmatrix} \cdot \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{bmatrix} = \begin{bmatrix} r_p^k \\ r_d^k \\ r_c^k \end{bmatrix} = \begin{bmatrix} b - A \cdot \tilde{x}^k \\ c - A^T \cdot \tilde{y}^k - \tilde{z}^k \\ \mu^k \cdot e - \Delta\tilde{X}^k \cdot \Delta\tilde{Z}^k \cdot e \end{bmatrix}. \tag{3.35}$$

A partir da nova direção, calcula-se o próximo ponto.

Observa-se que no método preditor-corretor são calculados dois sistemas não-lineares a cada iteração. No entanto, o custo de resolver os dois sistemas é atenuado porque é usado o mesmo Jacobiano.

O funcionamento do método pode ser resumido na seqüência de passos descrita a seguir.

A partir de um ponto inicial primal-dual interior ($\left(x^0, z^0\right) > 0$ e $y^\circ \in \Re^n$) e $\tau \in (0,1)$, para $k = 0,1,2,...,$ fazer até convergir:

1. Calcular a direção afim, $\left(\Delta\tilde{x}^k, \Delta\tilde{y}^k, \Delta\tilde{z}^k\right)$.
2. Calcular o tamanho dos passos primal e dual, de forma a manter os pontos auxiliares interiores, $\tilde{\alpha}_p^k = min\left(1, \tau \cdot \tilde{\rho}_p^k\right)$, $\tilde{\alpha}_p^k = min\left(1, \tau \cdot \tilde{\rho}_d^k\right)$, $\tilde{\rho}_p^k = \dfrac{-1}{\min\limits_j \left(\dfrac{\Delta\tilde{x}_i^k}{x_j^k} \right)}$ e $\tilde{\rho}_d^k = \dfrac{-1}{\min\limits_j \left(\dfrac{\Delta\tilde{z}_i^k}{z_j^k} \right)}$.
3. Calcular μ^k.
4. Calcular a nova direção, $\left(\Delta x^k, \Delta y^k, \Delta z^k\right)$.
5. Calcular o tamanho dos passos primal e dual, de forma a manter os novos pontos interiores, $\alpha_p^k = min\left(1, \tau \cdot \rho_p^k\right)$, $\alpha_p^k = min\left(1, \tau \cdot \rho_d^k\right)$, $\rho_p^k = \dfrac{-1}{\min\limits_j \left(\dfrac{\Delta x_i^k}{x_j^k} \right)}$ e $\rho_d^k = \dfrac{-1}{\min\limits_j \left(\dfrac{\Delta z_i^k}{z_j^k} \right)}$.
6. Calcular o novo ponto, $\left(x^{k+1}, y^{k+1}, z^{k+1}\right) = \left(x^k, y^k, z^k\right) + \alpha^k \cdot \left(\Delta x^k, \Delta y^k, \Delta z^k\right)$.

3.4 PONTO INICIAL

Nos métodos afins apresentados neste capítulo, é necessário o cálculo de um ponto inicial interior e factível para iniciar o processo de otimização. Nos métodos primais-duais, é suficiente que os pontos iniciais sejam interiores.

Para o cálculo do ponto inicial interior e factível para os métodos afins, é utilizada a inicialização clássica por FASE I [Adler *et al.* 1989; Tsuchiya 1996]. Usando este procedimento, o problema é resolvido em duas etapas. Na primeira etapa (FASE I), é calculado um ponto inicial interior factível, a partir da definição de um novo problema derivado do problema de otimização. Na segunda etapa (FASE II), o método afim escala

de pontos interiores realiza a otimização do problema a partir do ponto obtido na FASE I.

Este trabalho propõe um conjunto de possibilidades para obtenção de pontos iniciais para os métodos de pontos interiores através de redes de Hopfield. Uma das alternativas propostas procura avançar no processo de otimização com as redes de Hopfield, obtendo inicializações "a quente"[2] para os métodos de pontos interiores. A origem desse conceito é discutida brevemente na próxima seção.

3.5 INICIALIZAÇÃO A QUENTE EM MÉTODOS DE PONTOS INTERIORES

O cálculo de um ponto inicial avançado, ou inicialização "a quente", em métodos de pontos interiores é um problema pesquisado por vários autores [Gondzio 1996; Gondzio e Vial 1997; Yildirm e Wright 2002].

Normalmente, as abordagens para inicialização a quente procuram calcular um ponto inicial a partir de um "problema perturbado", que possui a mesma dimensão que o problema original, com pequenas perturbações em A, b e c. A motivação para esses estudos é a possibilidade de reduzir o número de iterações dos métodos e, conseqüentemente, o esforço computacional. Um aspecto a ser observado no cálculo desses pontos iniciais é procurar centrá-los, evitando a proximidade das fronteiras [Ross *et al.* 1997]. Caso contrário, o método de pontos interiores pode vir a realizar muitas iterações para se afastar da fronteira.

[2] Tradução livre do inglês *warm start*.

3.6 SOLUÇÃO DOS SISTEMAS LINEARES ORIUNDOS DOS MÉTODOS DE PONTOS INTERIORES

Como foi apresentado nas seções anteriores, todos os métodos de pontos interiores envolvem a resolução de sistemas lineares do tipo:

$$\left(A \cdot D^{-1} \cdot A^T\right) \cdot p = q\,, \tag{3.36}$$

A solução desses sistemas é o passo de maior esforço computacional dos métodos de pontos interiores. Nesta seção abordaremos a sua solução.

A matriz A da Equação (3.36) é um dado do problema e permanece constante ao longo de todas as iterações. D é uma matriz diagonal que varia a cada iteração, com características particulares para cada um dos métodos:

- $D = X^{-2}$ no método primal afim escala;
- $D = Z^2$ no método dual afim escala;
- $D = X^{-1} \cdot Z$ no método primal-dual e no método preditor-corretor.

A matriz $B = \left(A \cdot D^{-1} \cdot A^T\right)$ é simétrica e definida positiva. Ou seja,

- $B^T = B$ e
- $t^T \cdot \left(A \cdot D^{-1} \cdot A^T\right) \cdot t > 0, \quad \forall t \neq 0$ [Golub 1996].

3.6.1 Decomposição e Cálculo da Inversa

Devido às características da matriz B existe uma única matriz triangular inferior L com as características a seguir [Golub 1996]:

$$B = L \cdot L^T \tag{3.37}$$

onde os elementos da diagonal de L são estritamente positivos.

Esta fatoração, chamada de decomposição de *Cholesky*, é muito utilizada em método de pontos interiores.

Quando a inversa da matriz B é calculada explicitamente, são envolvidas muitas operações de ponto flutuante. A decomposição $L \cdot L^T$ permite reduzir este número. Utilizando-se a decomposição, o sistema (3.36) pode ser resolvido como:

$$\left(L \cdot L^T\right) \cdot p = q \tag{3.38}$$

$$L \cdot \left(L^T \cdot p\right) = q \tag{3.39}$$

$$L \cdot s = q \tag{3.40}$$

Uma das motivações para a combinação das técnicas de Redes de Hopfield e métodos de pontos interiores foi a semelhança entre os sistemas resolvidos para a solução de problemas de otimização linear por redes de Hopfield e os sistemas oriundos de pontos interiores.

Em redes de Hopfield para otimização linear é utilizada uma matriz de projeção definida como:

$$T = I - A^T \cdot \left(A \cdot A^T\right)^{-1} \cdot A \tag{3.41}$$

onde a matriz $C = A \cdot A^T$ possui uma estrutura similar à matriz B, com $D = I$.

Se pensarmos na possibilidade de calcular os pontos iniciais para métodos de pontos interiores por redes de Hopfield modificadas, todo o trabalho computacional realizado para o cálculo da decomposição da matriz T (da projeção) pode ser reutilizado no cálculo da decomposição da matriz B, dos métodos de pontos interiores.

Quando uma decomposição é calculada, realiza-se uma etapa preliminar, denominada fatoração simbólica [Duff *et al.* 1986]; nessa etapa, identificam-se os elementos não

nulos da matriz e procura-se reduzir o "enchimento"[3], através de permutações de linhas e colunas.

Quando redes de Hopfield são utilizadas para inicialização "a quente" de métodos de pontos interiores, a fatoração simbólica utilizada para a matriz *C*, das redes de Hopfield pode ser reutilizada para o cálculo da decomposição da matriz *B* dos métodos de pontos interiores; as matrizes possuem estruturas idênticas.

3.6.2 Esparsidade

Esparsidade é a relação entre o número de elementos não nulos e o número total de elementos da matriz [Duff *et al.* 1986]. Quando levamos em conta as esparsidades das matrizes *A* e *B*, podemos diminuir o número de operações no cálculo do sistema (3.38). Utilizando-se adequadamente estas informações, reduzem-se os espaços utilizados para armazenamento e os tempos de processamento, ao se evitar que sejam realizadas operações desnecessárias.

Podemos, por exemplo, utilizando-se o problema AFIRO e SCORPION do *Netlib*, realizar a operação $A \cdot x$, com utilização e não utilização das informações sobre esparsidade da matriz *A*. A Tabela 3.1 apresenta o número de operações em ponto flutuante para as duas situações.

TABELA 3.1 COMPARAÇÃO EM NÚMERO DE OPERAÇÕES QUANDO A ESPARSIDADE É UTILIZADA NAS OPERAÇÕES MATRICIAIS.

PROBLEMA	$(m \times n)$	**ESPASIDADE**	**NÚMERO DE OPERAÇÕES**	
			ESPARSA	**NÃO ESPARSA**
AFIRO	27x51	7.41%	848	15456
SCORPION	375x453	0.86%	3068	361616

[3] Do inglês "*fill in*", que significa o aparecimento de elementos não nulos ao longo do processo de solução do sistema, em posições onde existiam elementos nulos.

3.7 PRÉ-PROCESSAMENTO

Um aspecto importante nas implementações de métodos de pontos interiores é o pré-processamento [Gondzio 1997a], que corresponde à transformação do problema original em uma forma mais apropriada. O pré-processamento coloca o problema na forma padrão, elimina redundâncias e procura reduzir o tamanho do mesmo. Muitos problemas não podem ser resolvidos sem esta transformação prévia e, na maioria dos casos, o número total de iterações é reduzido.

O pré-processamento neste trabalho é realizado através do pré-processador do *LIPSOL* (Linear Programming Interior Point Solver v0.4) [Zhang 1998], (http://www.caam.rice.edu/~zhang/lipsol/). O *LIPSOL* é um software livre programado em Matlab para a solução de problemas de otimização linear de grande porte, através do método preditor-corretor.

A rotina de pré-processamento do *LIPSOL* realiza as seguintes operações:

1. Verifica a consistência dos limites das variáveis, observando se $\underline{x} \leq \overline{x}$ quando $\underline{x} \leq x \leq \overline{x}$.
2. Elimina variáveis fixas, isto é, se $\underline{x}_i = \overline{x}_i$ então $x_i = \overline{x}_i$.
3. Elimina linhas iguais a zero, verificando ao mesmo tempo a factibilidade.
4. Elimina colunas iguais a zero, verificando ao mesmo tempo se o problema é ilimitado.
5. Elimina linhas com uma única variável.

3.8 BIBLIOTECA *NETLIB*

Os estudos de casos realizados no Capítulo 5 e no Capítulo 6 utilizam um subconjunto de problemas de otimização linear da biblioteca *Netlib* (http://www.netlib.org). Esta biblioteca é amplamente utilizada por pesquisadores da área de otimização [Gay 1985].

CAPÍTULO 4.

REDES DE HOPFIELD MODIFICADAS

Este capítulo apresenta as redes de Hopfield modificadas, propostas por Silva [1997], que motivaram o desenvolvimento das abordagens cooperativas com métodos de pontos interiores, objeto deste trabalho. Apresenta também os aperfeiçoamentos introduzidos nas redes de Hopfield modificadas, inspirados nos algoritmos de pontos interiores e em resultados recentes na área de álgebra matricial. No final do capítulo, discutem-se as motivações para o desenvolvimento das abordagens cooperativas.

As Redes de Hopfield modificadas foram definidas para a solução de problemas de otimização linear, não linear [Silva *et al.* 1998], quadrática, dinâmica [Silva *et al.* 1999] e problemas de identificação robusta [Silva *et al.* 1997].

4.1 SOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO POR REDES DE HOPFIELD

O primeiro trabalho para a resolução de problemas de otimização através de redes neurais foi proposto por Tank e Hopfield [1986]. Nesta rede, definida para problemas lineares com restrições lineares, é construída uma função de energia a partir da função objetivo do problema de otimização e das restrições, que são incorporadas como termos de penalidade. A rede é definida em um ciclo fechado de forma que, quando uma restrição é violada, o valor da violação é realimentado na rede.

Deve-se destacar que um aspecto frágil na abordagem desenvolvida por Tank e Hopfield [1986] é o fato de que o ponto de equilíbrio não corresponde a uma solução ótima do problema original porque não satisfaz as condições de otimalidade de Kuhn-Tucker [Maa e Shanblatt 1992].

As redes de Hopfield modificadas propostas por Silva [1997] resolvem o problema de otimização em duas etapas. Na primeira etapa, denominada fase de factibilização, o ponto é factibilizado, fazendo com que todas as restrições do problema sejam satisfeitas; na segunda etapa, denominada fase de atualização, calcula-se o próximo ponto, a partir do ponto factível e da direção de busca. Com esta nova metodologia, as restrições não precisam ser incorporadas como termos de penalidade na função de energia, pois a factibilidade é garantida.

4.2 SOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO POR REDES DE HOPFIELD MODIFICADAS

As redes de Hopfield modificadas constituem uma metodologia geral para resolver problemas de otimização do tipo caracterizado a seguir:

$$\begin{aligned} &min \quad f(x) \\ &sujeito\ \ a \quad g(x) = 0 \\ &\qquad\quad\ \ h(x) \le 0 \\ &\underline{x} \le x \le \overline{x} \end{aligned} \tag{4.1}$$

onde,

$x \in \Re^n$ é o vetor de variáveis do problema,

$f : \Re^n \to \Re$ é a função objetivo,

$g : \Re^n \to \Re^p$, $h : \Re^n \to \Re^q$ são as restrições que definem a região de factibilidade e

$\underline{x}, \overline{x} \in \Re^n$ representam os limitantes inferiores e superiores do vetor *x*.

Silva [1997] mostra que a solução do problema (4.1) pode ser encontrada por um processo realizado em duas fases: fase de factibilização e fase de atualização. Em cada fase é realizada a minimização de uma função de energia. Na fase de factibilização, é minimizada a função de energia (4.2). Na fase de atualização, é minimizada a função de energia (4.3).

$$E_{rest} = -\frac{1}{2} \cdot x^T \cdot T \cdot x - x^T \cdot p_r , \tag{4.2}$$

$$E_{obj} = -\frac{1}{2} \cdot x^T \cdot Q \cdot x - x^T \cdot p_o . \tag{4.3}$$

onde a matriz T e o vetor p_r estão associados às restrições do problema de otimização e a matriz Q e o vetor p_o são determinados a partir da função objetivo do problema [Silva 1997].

A solução que minimiza simultaneamente as funções E_{rest} e E_{obj} corresponde à solução ótima do problema de otimização.

A Fig. 4.1 ilustra a metodologia proposta por Silva, que pode ser resumida na seqüência de passos a seguir:

1. Calcular um ponto inicial x_0.
2. Enquanto a energia não for estável:
   I. Fase de atualização, representada pelo bloco II da Fig. 4.1;
   II. Fase de factibilização, representada pelo bloco I da Fig. 4.1.

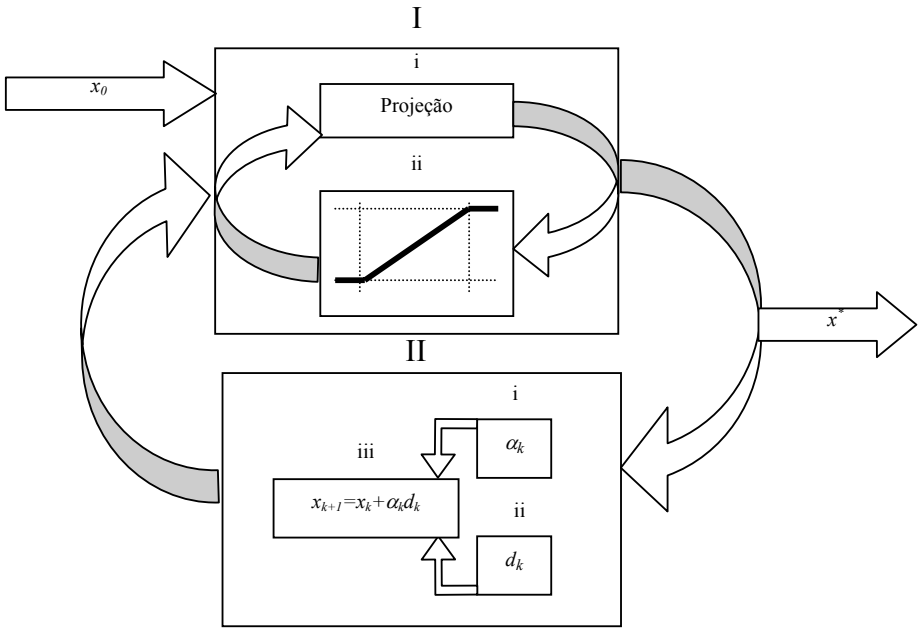

Fig. 4.1. Redes de Hopfield Modificadas

4.2.1 Fase de Factibilização

Nesta fase, o ponto x_k é transformado em um ponto factível, dentro do espaço definido pelas restrições do problema de otimização. A obtenção do ponto factível é realizada em dois passos, repetidos até que a função de energia seja estabilizada.

1. Projeção no subespaço válido.[Aiyer *et al.* 1990; Aiyer e Fallside 1991]

O novo ponto, x_p^k, é obtido a partir da Equação (4.4):

$$x_p^k = T \cdot x^k + s, \tag{4.4}$$

onde *T* é a matriz de projeção que representa as conexões entre os neurônios (Fig. 4.2) e *s* é o *bias*, calculado como ortogonal a *T*. A matriz *T* e o vetor *s* são calculados a partir das restrições do problema de otimização [Aiyer *et al.* 1990; Aiyer e Fallside 1991].

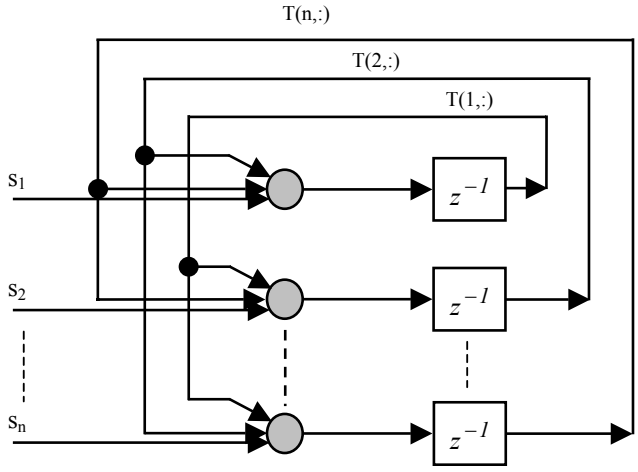

Fig. 4.2. Redes de Hopfield da Fase de Factibilização

2. Aplicação da função de ativação.

A função de ativação, $g(x)$, obtém um novo ponto no hipercubo definido pelas restrições de canalização. Este passo é representado pela caixa I.ii de Fig. 4.1. A função $g(x)$ é definida da forma a seguir.

$$g(x) = [g_1(x), g_2(x), \ldots, g_n(x)]^T \tag{4.5}$$

$$g_i(x_i) = \begin{cases} \underline{x}_i & x_i \le \underline{x}_i \\ x_i & \underline{x}_i < x_i < \overline{x}_i \\ \overline{x}_i & x_i \ge \overline{x}_i \end{cases}, \tag{4.6}$$

onde $\underline{x}_i$ e $\overline{x}_i$ são o limitante inferior e superior do elemento x_i, definidos no problema (4.1).

Em resumo, quando um ponto entra na fase de factibilização, inicialmente faz-se sua projeção no subespaço válido. Porém, o ponto obtido com a projeção pode não satisfazer as restrições de canalização do problema; a aplicação da função de ativação obtém um novo ponto, que pode violar uma ou mais entre as demais restrições. Assim, realiza-se novamente a projeção, repetindo-se o processo até que o ponto seja totalmente factível. Esta condição é satisfeita em um ponto de equilíbrio da função de energia E_{rest}.

Observa-se que os aspectos principais do procedimento proposto por Silva para a obtenção de um ponto factível em um problema de otimização são o cálculo da matriz *T*, do vetor *s* e da função $g(x)$, a partir das restrições do problema.

4.2.2 Fase de Atualização

Na fase de atualização, é realizada a minimização da função de energia E_{obj} definida a partir da função objetivo do problema de otimização. O próximo ponto x^{k+1} é obtido em uma direção de busca d^k.

A direção de busca d^k, nesta metodologia, é calculada como oposta ao gradiente da função E_{obj}, em relação a x^k, isto é:

$$d^k = -\nabla E(x) = Q \cdot x^k + p_o \tag{4.7}$$

Por outro lado, o passo α_k é calculado por um controlador nebuloso [Pedrycz e Gomide 1998] que utiliza a variação da função de energia e informações sobre a distância entre pontos anteriores [Silva 1997].

Em resumo, a atualização do ponto x^k é realizada com a seqüência de passos a seguir.

1. Cálculo da direção, d^k, pela Equação (4.7).

2. Cálculo do passo α^k, pelo controlador nebuloso.

3. Cálculo do novo vetor x^{k+1} pela expressão (4.8).

$$x^{k+1} = x^k + \alpha^k \cdot d^k \tag{4.8}$$

Quando o ponto x^{k+1} é calculado na fase de atualização, o ponto pode ser infactível, porque a forma de cálculo da direção a partir da função objetivo do problema e do passo definido pelo controlador nebuloso não levam em conta a região de factibilidade do problema. Conseqüentemente, o novo ponto deve passar pela fase de factibilidade. Este processo de duas fases é repetido até que a função de energia E_{obj} seja estabilizada.

4.2.3 Redes de Hopfield Modificadas para Resolver Problemas Lineares

Deseja-se abordar, através de redes de Hopfield modificadas, problemas de otimização linear na forma padrão, apresentada a seguir:

$$\begin{aligned} &min \quad c^T \cdot x \\ &sujeito \;\; a \;\; A \cdot x = b \\ &\qquad 0 \le x \le \bar{x} \end{aligned} \tag{4.9}$$

Silva mostra que, neste caso, a matriz *Q* é nula, fazendo com que a função de energia E_{obj} coincida com a própria função objetivo do problema e as direções d^k, calculadas pela Equação (4.10) sejam sempre identificadas com o negativo do vetor de custos.

$$d = -\nabla E(x) = Q \cdot x - c = -c \tag{4.10}$$

As equações para a matriz *T* e o vetor *s* da equação da projeção do problema (4.9) são definidas como segue [Silva 1997]:

$$T = I - A^T \cdot \left(A \cdot A^T\right)^{-1} \cdot A \tag{4.11}$$

$$s = A^T \cdot \left(A \cdot A^T\right)^{-1} \cdot b\,. \tag{4.12}$$

onde I é a matriz identidade.

Pode ser verificado que a matriz de projeção T e o vetor s satisfazem as propriedades de simetria, idempotência e ortogonalidade, exigidas para a solução do problema.

1. Simetria, i. e., $T = T^T$.

$$\begin{aligned} T^T &= \left(I - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \right)^T = I^T - \left(A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \right)^T = \\ &= I - A^T \cdot \left(A^T \cdot \left(A \cdot A^T \right)^{-1} \right)^T = I - A^T \cdot \left(A \cdot A^T \right)^{-I^T} \cdot A = I - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \\ &= T \end{aligned}$$

2. Idempotência, i.e., $T \cdot T = T$.

$$\begin{aligned} T \cdot T &= \left(I - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \right) \cdot \left(I - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \right) \\ &= I - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A + A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \cdot A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \\ &= I - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A + A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \\ &= I - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \\ &= T \end{aligned}$$

3. Ortogonalidade do vetor s em relação à matriz T, i.e., $T \cdot s = 0$

$$\begin{aligned} T \cdot s &= \left(I - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \right) \cdot \left(A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot b \right) \\ &= A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot b - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \cdot A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot b \\ &= A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot b - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot b \\ &= 0 \end{aligned}$$

A partir das expressões para T e s (Equação 4.11, Equação 4.12), é fácil mostrar que o ponto obtido é factível. De fato, considere as restrições do problema:

$$A \cdot x = b \tag{4.13}$$

e a equação de projeção,

$$x = T \cdot x + s \,. \tag{4.14}$$

Usando-se as equações (4.11) e (4.12), tem-se:

$$\begin{aligned} A \cdot (T \cdot x + s) &= A\left(I - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \right) x + A \cdot A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot b \\ &= A \cdot x - A \cdot A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \cdot x + b \\ &= A \cdot x - A \cdot x + b \\ &= b \end{aligned} \tag{4.15}$$

Logo, $A \cdot x = b$.

A função de ativação $g(x)$ para o problema (4.9) é definida pela Equação (4.16).

$$g(x) = \begin{cases} 0 & x \le 0 \\ x & 0 < x < \bar{x} \\ \bar{x} & x \ge \bar{x} \end{cases}, \tag{4.16}$$

4.3 APERFEIÇOAMENTOS NAS REDES DE HOPFIELD MODIFICADAS

Nas redes de Hopfield modificadas, foram realizados os seguintes aperfeiçoamentos ou modificações para a solução de problemas lineares de otimização.

1. Na projeção, foram utilizadas técnicas para a solução de sistemas lineares oriundos de métodos de pontos interiores, diminuindo o número de operações em ponto flutuante e permitindo a sua aplicação em problemas reais de otimização.

2. Foi modificada a função de ativação para obter pontos interiores—esta modificação foi necessária para a utilização de direções de pontos interiores;

3. Foram utilizadas as direções dos métodos de pontos interiores na fase de atualização—a direção fixa $d = c$ foi substituída.

4. Foi utilizado um passo fixo na fase de atualização (o controlador nebuloso não é apropriado para as novas direções introduzidas).

Nas seções que seguem, serão discutidos estes aperfeiçoamentos.

4.3.1 Modificações na Projeção

A matriz *T* e o vetor *s* utilizados na equação de projeção do vetor *x* no subespaço válido são calculados pelas expressões (4.11) e (4.12), para um problema de otimização linear. Quando estas equações são utilizadas diretamente nas implementações, muitas operações desnecessárias em ponto flutuante são realizadas; mesmo em situações em que se procura utilizar esparsidade da matriz *A*.

Utilizando as equações (4.11) e (4.12) o vetor de projeção x_p é calculado como:

$$x_p = \left(I - A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot A \right) \cdot x + A^T \cdot \left(A \cdot A^T \right)^{-1} \cdot b . \tag{4.17}$$

Logo,

$$x_p = x + \left(A^T \cdot \left(A \cdot A^T \right)^{-1} \right) \cdot \left(b - A \cdot x \right). \tag{4.18}$$

A matriz $A \cdot A^T$ é calculada uma única vez em um passo prévio de inicialização antes do processo de convergência das redes. Como $A \cdot A^T$ é uma matriz definida positiva, pode-se utilizar a decomposição de *Cholesky* [Golub 1996] para o manuseio desta matriz na Equação (4.18). Assim,

$$A \cdot A^T = L_R \cdot L_R^T . \tag{4.19}$$

A matriz L_R pode ser utilizada na equação de projeção (4.18), como segue:

$$x_p = x + A^T \cdot \left(L_R{}^{-1} \cdot \left(L_R^{T\,-1} \cdot \left(b - A \cdot x \right) \right) \right) \tag{4.20}$$

O cálculo do vetor de projeção pela Equação (4.20) normalmente diminui significativamente o número de operações em ponto flutuante.

4.3.2 Modificações na Função de Ativação

Uma das modificações propostas para estas redes é a utilização das direções dos métodos de pontos interiores. No cálculo destas direções, são utilizados pontos interiores

à região de factibilidade. Conseqüentemente, o ponto obtido na fase de factibilização precisa ser interior.

Para obter um ponto interior pelas redes de Hopfield modificadas, é necessário realizar uma pequena modificação na função de ativação. Para isso, utiliza-se um parâmetro chamado de interioridade (*It*) na função $g(x)$ da Equação (4.16). Este parâmetro determina a distância mínima de todas as componentes do vetor *x* à fronteira da região de factibilidade para o problema (4.9). A nova função de ativação $g_{it}(x)$ é apresentada na Equação (4.21).

$$g_{it}(x) = \begin{cases} It & x \le It \\ x & It < x < \overline{x} - It \\ \overline{x} - It & x \ge \overline{x} - It \end{cases} \tag{4.21}$$

Por exemplo, em um problema em que existem restrições do tipo $A_u \cdot x \ge b_u$ ou $A_l \cdot x \le b_l$, a nova região de factibilidade criada por este parâmetro é ilustrada na Fig. 4.3.

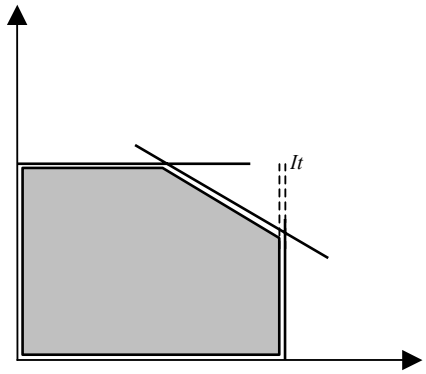

Fig. 4.3. Nova Região de Factibilidade Definida pela Interioridade *It*.

4.3.3 Nova Direção de Busca

Pode-se aperfeiçoar a direção de busca, definida por Silva [1997] para problemas lineares, utilizando-se conceitos de métodos de pontos interiores.

Os métodos de pontos interiores apresentados no capítulo anterior podem proporcionar direções mais ricas, com movimentações por dentro da região de factibilidade. Exploram-se direções derivadas dos métodos afins e primais-duais.

4.3.4 O Passo da Direção de Busca

Silva propôs um controlador baseado em álgebra nebulosa para o cálculo do passo na fase de atualização das redes de Hopfield modificadas. Com a substituição das direções fixas por direções de métodos de pontos interiores, o passo calculado pelo controlador nebuloso não é o mais apropriado.

Utilizou-se um passo fixo, $\alpha = 1$; pode-se argumentar que esta é a melhor alternativa quando se explora a cooperação de redes de Hopfield com métodos primais-duais.

4.4 MOTIVAÇÃO PARA NOVAS ABORDAGENS

Silva [1997] exemplifica a convergência das redes de Hopfield modificadas para solução de problemas de otimização linear, através de um problema de *6* variáveis e *4* restrições. A Fig. 4.4[1] ilustra o processo de convergência da rede para este problema.

A primeira motivação para o uso de redes de Hopfield em inicializações avançadas foi o desempenho destas redes na solução deste problema. Pode-se verificar que a rede tem uma convergência acelerada nas primeiras iterações, mas depois o processo de otimização vai diminuindo a sua velocidade.

---

[1] Esta figura faz parte da Tese de Doutorado de Ivan Nunes da Silva [1997]

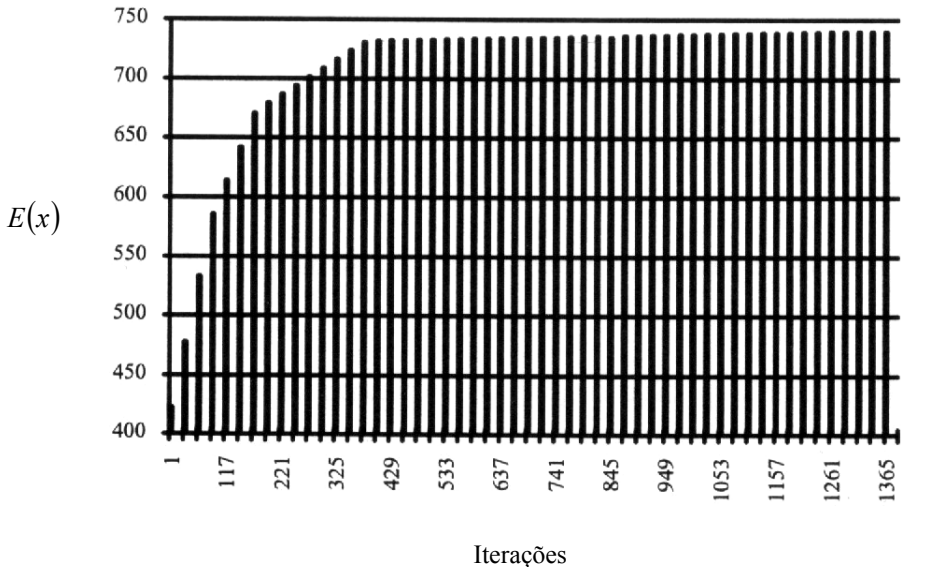

Fig.4.4. Comportamento da Função Objetivo para o Problema de Silva [1997]

Os aperfeiçoamentos introduzidos neste capítulo permitiram resolver sistemas maiores. Vamos então verificar o comportamento da nova abordagem no problema AFIRO da biblioteca *Netlib* (a matriz *A* deste problema tem *27* linhas e *51* colunas). Na Fig. 4.5 são ilustrados os resultados obtidos pelas redes utilizando a direção primal afim escala para a solução do problema.

No comportamento das redes de Hopfield com os aperfeiçoamentos propostos, pode-se verificar novamente um comportamento análogo:

- Convergência rápida nas primeiras iterações. Por exemplo, na iteração $I = 4$, o valor da função de energia foi $E_{obj} = -455.9183$ (o valor da função de energia no ótimo do problema é $E^*_{obj} = -464.7531$);

- Convergência muito lenta no final do processo de otimização. Na iteração $I = 100$, o valor da função objetivo não mudou muito em relação à iteração $I = 4$.

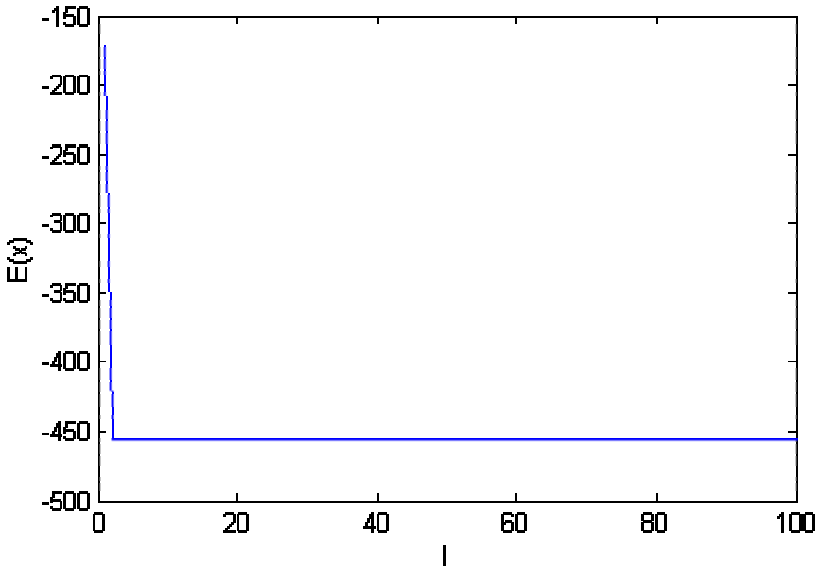

Fig. 4.5. Convergência das Redes de Hopfield Modificadas para o Problema AFIRO.

A melhora rápida da função objetivo no início do processo de otimização nos levou a pensar na possibilidade da utilização desta abordagem para inicializações "a quente" de métodos de pontos interiores. Isto é, utilizar as redes de Hopfield modificadas, com os aperfeiçoamentos discutidos, para o cálculo de pontos iniciais avançados para métodos de pontos interiores, procurando diminuir o número total de iterações no processo de otimização.

Outro aspecto que "convida" a colaboração de redes de Hopfield modificadas com os métodos de pontos interiores é a semelhança entre os sistemas de equações lineares adotados nas duas abordagens.

Como foi apresentado no capítulo anterior, os sistemas utilizados na projeção das variáveis para obter um ponto factível e os sistemas lineares resolvidos no cálculo das direções em métodos de pontos interiores (Equação (4.22)) possuem estruturas idênticas.

$$\left(A \cdot D^{-1} \cdot A^{T}\right) \cdot p = q \tag{4.22}$$

Esta similaridade na estrutura permite melhorar a eficiência no processo de convergência do método de pontos interiores. O trabalho computacional de fatoração simbólica [Duff *et al.* 1986] realizado para o cálculo da decomposição das matrizes de projeção $A \cdot A^T$ pode ser re-utilizado nos sistemas lineares dos métodos de pontos interiores (Equação 4.22), para o cálculo das direções de busca.

CAPÍTULO 5.

OTIMIZAÇÃO POR REDES NEURAIS E MÉTODOS AFINS ESCALA DE PONTOS INTERIORES

A cooperação entre redes neurais e métodos de pontos interiores pode ser realizada em dois sentidos: métodos de pontos interiores para o treinamento de redes neurais [Trafalis *et al.* 1994; 1996; 1997; Lemmon e Szymanski 1994; Szymanski *et al.* 1998] e ambas as técnicas trabalhando juntas para resolver problemas de otimização [Velazco *et al.* 2002a; 2002b; 2003].

Este capítulo explora as possibilidades de cooperação das técnicas de Redes Neurais e métodos afins escala de pontos interiores para solução de problemas de otimização linear. As abordagens propostas utilizam as redes de Hopfield modificadas [Silva 1997], com aperfeiçoamento nas direções de busca e nos métodos de manuseio das matrizes,

para obter inicializações avançadas para métodos de pontos interiores. Para avaliação das abordagens, realizam-se estudos de casos com problemas reais de programação linear da biblioteca *Netlib*.

5.1 Abordagens Alternativas para a Inicialização dos Métodos Afins Escala

Para a inicialização dos métodos afins escala foram desenvolvidas quatro abordagens: duas abordagens para o método primal afim escala e duas abordagens para o método dual afim escala. As operações com redes de Hopfield modificadas utilizam os aperfeiçoamentos discutidos no capítulo anterior.

As quatro abordagens podem ser resumidas da seguinte forma:

1. Obtenção de pontos iniciais interiores factíveis para o método primal afim escala através da fase de factibilização das redes de Hopfield modificadas;
2. Obtenção de pontos interiores factíveis avançados para o método primal afim escala através de redes de Hopfield modificadas com a direção primal afim escala—denomina-se esta abordagem de Hopfield-primal;
3. Obtenção de pontos iniciais interiores factíveis para o método dual afim escala através da fase de factibilização das redes de Hopfield modificadas;
4. Obtenção de pontos interiores factíveis avançados para o método dual afim escala através de redes de Hopfield modificadas com a direção dual afim escala—denomina-se esta abordagem de Hopfield-dual.

5.1.1 Inicialização para o Método Primal Afim Escala

Nesta seção, são apresentadas as abordagens para a inicialização avançada do método primal afim escala.

Para uma melhor compreensão dos métodos, deve-se relembrar que um problema de otimização linear pode ser apresentado na forma padrão do problema primal como segue:

$$\begin{array}{c} min \quad c^T \cdot x \\ sujeito \;\; a \;\; A \cdot x = b, \\ x \geq 0 \end{array} \tag{5.1}$$

A. Inicialização pelo processo de factibilização das redes de Hopfield modificadas

O processo de obtenção de um ponto interior factível ou processo de factibilização para o primal, pode ser resumido nos passos a seguir.

I. Projeção do ponto *x* no subespaço válido utilizando a Equação (5.2).

$$x_p = x + A^T \cdot \left(L_R^{-1} \cdot \left(L_R^{T\,-1} \cdot (b - A \cdot x) \right) \right) \tag{5.2}$$

A matriz L_R é obtida pela decomposição de *Cholesky* da matriz $A \cdot A^T$.

$$A \cdot A^T = L_R \cdot L_R^T. \tag{5.3}$$

II. Aplicação da função de ativação (5.4).

$$g_{it}(x) = \begin{cases} It & x \leq It \\ x & It < x \end{cases}. \tag{5.4}$$

Como descrito no Capítulo 4, o parâmetro *It*, chamado de interioridade, garante que o ponto seja interior à região de factibilidade.

Os passos I e II são repetidos até que o ponto *x* seja interior e factível, isto é:

$$\begin{array}{l} A \cdot x = b \\ x \geq It \end{array}. \tag{5.5}$$

B. Inicialização pelo método Hopfield-Primal

O método Hopfield-primal calcula um ponto interior factível avançado, para o método primal afim escala. O processo pode ser resumido na seqüência de passos a seguir.

I. Calcular um ponto primal inicial factível x_0, pelo processo de factibilização, descrito na seção anterior.

II. Aplicar o processo de atualização-avançada-primal para calcular o próximo ponto:

i. Calcular a direção primal afim escala, Δx;

ii. Calcular o próximo ponto, $x^{k+1} = x^k + \alpha \cdot \Delta x$.

III. Calcular um novo ponto primal factível a partir do ponto x^{k+1}, pelo processo de factibilização.

IV. Se o ponto x^{k+1} for considerado "suficientemente avançado", utilize o método primal afim escala para conclusão da otimização; caso contrário, incremente o contador *k* e volte ao Passo II.

Uma métrica para a verificação de quanto o novo ponto factível esta "avançado" no espaço de soluções pode ser construída a partir da melhora na função objetivo. No entanto, o estudo de casos sugere que uma alternativa simples para se obter um ponto "suficientemente avançado" é através de adoção de um número fixo de iterações.

5.1.2 Inicialização para o Método Dual Afim Escala

A partir do problema primal (5.1) pode-se formular o problema dual, na forma a seguir:

$$\begin{aligned} max \quad & b^T \cdot y \\ sujeito\ a \quad & A^T \cdot y + z = c \\ & z \geq 0 \end{aligned} \tag{5.6}$$

Este problema pode ser re-escrito como:

$$\begin{aligned} max \quad & b^T \cdot y \\ sujeito\ a \quad & \begin{bmatrix} A^T & I_n \end{bmatrix} \cdot \begin{bmatrix} y \\ z \end{bmatrix} = c, \\ & z \geq 0 \end{aligned} \tag{5.7}$$

onde I_n é a matriz identidade $n \times n$.

A. Inicialização pelo processo de factibilização para o dual

O processo de factibilização para o problema primal pode ser facilmente estendido para o problema dual (5.7).

A matriz T e o vetor s da equação de projeção podem ser reformulados para o problema dual, como segue:

$$T = I_{m+n} - \begin{bmatrix} A^T & I_n \end{bmatrix}^T \cdot \left(\begin{bmatrix} A^T & I_n \end{bmatrix} \cdot \begin{bmatrix} A^T & I_n \end{bmatrix}^T \right)^{-1} \cdot \begin{bmatrix} A^T & I_n \end{bmatrix}, \tag{5.8}$$

$$s = \begin{bmatrix} A^T & I_n \end{bmatrix} \cdot \left(\begin{bmatrix} A^T & I_n \end{bmatrix} \cdot \begin{bmatrix} A^T & I_n \end{bmatrix}^T \right)^{-1} \cdot c\,. \tag{5.9}$$

Logo, as equações de projeção das variáveis y e z são:

$$y_p = y - A \cdot \left(A^T \cdot A + I_n \right)^{-1} \cdot \left(A^T \cdot y + z - c \right), \tag{5.10}$$

$$z_p = z - \left(A^T \cdot A + I_n \right)^{-1} \cdot \left(A^T \cdot y + z - c \right). \tag{5.11}$$

Das equações (5.10) e (5.11), pode ser facilmente verificado que o ponto $\left(y_p, z_p \right)$ é factível.

Para isto, basta verificar que

$$A^T \cdot y_p + z_p = c\,. \tag{5.12}$$

Utilizando-se as equações de projeção do ponto $\left(y, z \right)$ e a Equação (5.12), tem-se:

$$A^T \cdot \left(y - A \cdot \left(A^T \cdot A + I_n \right)^{-1} \cdot \left(A^T \cdot y + z - c \right) \right) + z - \left(A^T \cdot A + I_n \right)^{-1} \cdot \left(A^T \cdot y + z - c \right) =$$

$$A^T \cdot y + z - \left(A^T \cdot A \right) \cdot \left(A^T \cdot A + I_n \right)^{-1} \cdot \left(A^T \cdot y + z - c \right) - \left(A^T \cdot A + I_n \right)^{-1} \cdot \left(A^T \cdot y + z - c \right) =$$

$$A^T \cdot y + z - \left(A^T \cdot A + I_n \right) \cdot \left(A^T \cdot A + I_n \right)^{-1} \cdot \left(A^T \cdot y + z - c \right) =$$

$$A^T \cdot y + z - \left(A^T \cdot y + z - c \right) = c$$

Logo,

$$A^T \cdot y_p + z_p = c \tag{5.13}$$

e o ponto (y_p, z_p) é factível.

Se utilizamos as equações (5.10) e (5.11) para a projeção do ponto (y,z) no subespaço válido, a cada ocorrência do processo de factibilização para o problema dual na rede de Hopfield, utiliza-se a matriz $\left(A^T \cdot A + I_n\right)^{-1}$ de dimensão $n \times n$. O uso desta matriz neste processo pode ser muito custoso, pois geralmente trabalhamos com problemas muito grandes.

Uma forma de melhorar o desempenho do processo de projeção é com a utilização de matrizes de dimensões menores, por exemplo $m \times m$ (deve-se lembrar que em um problema de otimização linear $m < n$). Isto pode ser viabilizado com a utilização da expressão de *Sherman-Morrison-Woodbury* [Golub 1996], definida na Equação (5.14).

$$\left(A^T \cdot A + I_n\right)^{-1} = I_n - A^T \cdot \left(A \cdot A^T + I_m\right)^{-1} \cdot A \tag{5.14}$$

A matriz $A \cdot A^T + I_m$ desta equação é definida positiva. Logo, pode-se utilizar a decomposição de *Cholesky* [Golub 1996] para o seu manuseio na equação de projeção.

$$A \cdot A^T + I_m = L_d \cdot L_d^T . \tag{5.15}$$

Quando a matriz $I_n - A^T \cdot \left(A \cdot A^T + I_m\right)^{-1} \cdot A$ é utilizada no cálculo do vetor de projeção (y_p, z_p), a cada ocorrência do processo de factibilização será utilizada uma matriz triangular esparsa $m \times m$; conseqüentemente, de dimensão menor do que a matriz $n \times n$, que seria utilizada sem a adoção deste aperfeiçoamento.

Com esta transformação, o vetor de projeção (y_p, z_p) pode ser calculado da forma a seguir.

$$y_p = y - A \cdot \left(I_n - A^T \cdot \left(L_d^{-1} \cdot L_d^{T^{-1}} \right) \cdot A \right) \cdot \left(A^T \cdot y + z - c \right) \tag{5.16}$$

$$z_p = z - \left(I_n - A^T \cdot \left(L_d^{-1} \cdot L_d^{T^{-1}} \right) \cdot A \right) \cdot \left(A^T \cdot y + z - c \right) \tag{5.17}$$

A função de ativação para a obtenção de pontos interiores pode também ser estendida para o problema dual. Para isto, é introduzido o parâmetro de interioridade It_d que determina a distância mínima dos elementos do vetor de variáveis *z* à fronteira (*y* é uma variável livre). A nova função de ativação é definida como segue:

$$g_{it_d}(y,z) = \begin{cases} (y, It_d) & z \le It_d \\ (y,z) & It_d < z \end{cases} \tag{5.18}$$

A abordagem pode ser resumida nos passos a seguir:

I. Projeção do ponto, utilizando-se as Equações (5.16) e (5.17);

II. Aplicação da função de ativação (5.18).

Os passos I e II são repetidos até que o ponto (y,z) seja interior e factível, isto é:

$$\begin{matrix} A^T \cdot y + z = c \\ z > It_d \end{matrix} . \tag{5.19}$$

B. Inicialização pelo método Hopfield-Dual

O cálculo do ponto dual avançado pelo método Hopfield-dual pode ser resumido nos passos a seguir.

I. Calcular um ponto dual inicial factível (y_0, z_0) pelo processo de factibilização para o dual.

II. Aplicar o processo de atualização-avançada-dual para calcular o próximo ponto:

i. Calcular a direção dual afim escala, $(\Delta y, \Delta z)$;

ii. Calcular o próximo ponto, $\left(y^{k+1}, z^{k+1}\right) = \left(y^k, z^k\right) + \alpha \cdot (\Delta y, \Delta z)$.

III. Calcular um novo ponto dual factível a partir do ponto $\left(y^{k+1}, z^{k+1}\right)$, pelo processo de factibilização para o dual.

IV. Se o ponto $\left(y^{k+1}, z^{k+1}\right)$ for considerado "suficientemente avançado", utilize o método dual afim escala para conclusão da otimização; caso contrário, incremente o contador *k* e volte ao Passo II.

De forma análoga ao método Hopfield-primal, uma métrica para a verificação de quanto o novo ponto factível esta "avançado" no espaço de soluções pode ser construída a partir da melhora na função objetivo. Mas, verifica-se também no estudo de casos que uma alternativa simples para se obter um ponto "suficientemente avançado" é através da adoção de um número fixo de iterações.

5.2 CÁLCULO DO PASSO.

Os métodos afins escala de pontos interiores apresentados no Capítulo 3 utilizam uma direção factível Δ para o cálculo do próximo ponto e adotam um passo α que mantém o ponto interior à região de factibilidade.

Nos métodos de inicialização Hopfield-primal e Hopfield-dual utiliza-se a mesma direção factível Δ. No entanto, é o processo de factibilização que se ocupa da interioridade do ponto a cada iteração. Utilizando-se esta metodologia não existe um limite superior para o passo que pode ser dado na direção factível (em métodos afim escala o limite é determinado pela fronteira da região).

Se utilizássemos o passo clássico dos métodos afins nas inicializações avançadas o próximo ponto obtido seria interior e factível, fazendo com que o processo de factibilização perdesse sua função. De esta forma, seriam realizadas apenas as próprias iterações dos métodos afins e perderíamos uma boa característica do processo de factibilização; especificamente, a de nos permitir obter pontos centrados na região de factibilidade.

Neste trabalho, foi utilizado o passo fixo $\alpha = 1$ (passo de Newton). Foram realizados testes com $\alpha < 1$ e $\alpha > 1$; em alguns casos, o número total de iterações foi menor, mas os estudos de casos não permitiram extrair informações conclusivas sobre os tamanhos de passos mais adequados.

5.3 ESTUDO DE CASOS

Esta seção apresenta o estudo de casos realizado com os métodos afins escala utilizando três inicializações: inicialização clássica por FASE I, inicialização pelo processo de factibilização de Hopfield e inicialização por redes de Hopfield modificadas, com direções afins.

O desempenho dos métodos com as diferentes inicializações foi avaliado em relação ao número total de iterações e ao tempo na convergência. Para os testes, foi utilizado um subconjunto de problemas do *Netlib*.

A maior parte dos estudos de casos foram realizados em um computador PC, Pentium IV, 2.26 GHz, 512 Mb de memória com sistema operacional Windows 2000. Os problemas mais difíceis foram realizados em outro computador PC, Pentium IV, 2.20 GHz, 1 Gb de memória, com sistema operacional Windows 2000. Deve-se observar que as varias abordagens para um determinado problema foram avaliadas no mesmo equipamento. Todos os métodos foram implementados na linguagem *Matlab 6.1*.

5.3.1 Problemas do Netlib Utilizados

A Tabela 5.1 apresenta o subconjunto de problemas da biblioteca *Netlib* escolhidos para o estudo de casos. A primeira coluna apresenta o nome do problema; a segunda coluna, o número de restrições m e de variáveis n de cada problema, após o pré-processamento inicial discutido na seção 3.7; a última coluna apresenta o valor das funções objetivos nas soluções ótimas (todos os problemas são de minimização), fornecidos por Bixby [1992].

TABELA 5.1. PROBLEMAS DO *NETLIB* UTILIZADOS

PROBLEMA	$(m \times n)$	**VALOR DO ÓTIMO**
AFIRO	27×51	-464.7531
SC50B	48×76	-70.0000
SC50A	49×77	-64.5750
SC105	104×162	-52.2020
ADLITTLE	55×137	225494.9631
SCAGR7	128×184	-2331389.2548
STOCFOR1	109×157	-41131.9762
BLEND	74×114	-30.8121
SC205	203×315	-52.2020
SHARE2B	96×162	-415.7322
LOTFI	151×364	-25.2647
SHARE1B	112×248	-76589.3185
SCAGR25	470×670	-14753433.061
SCTAP1	300×660	1412.2500
BRANDY	149×259	1518.5098
ISRAEL	174×316	-896644.8218
SCSD1	77×760	8.6666
AGG	488×615	-35991767.287
BANDM	269×436	-158.6280
E226	220×469	-18.7519
BEACONFD	148×270	33592.4858
SCSD6	147×1350	50.5000
SCFXM2	644×1184	36660.2615
FFFFF800	501×1005	555679.6116
SHIP12S	466×2293	1489236.1344
STOCFOR2	2157×3045	-39024.4085
SCSD8	397×2750	904.9999
SCTAP3	1480×3340	142.4000
25FV47	798×1854	5501.8458
SHIP121	838×5329	1470187.9193
TRUSS	1000×8806	458815.8471
D2Q06C	2171×5831	122784.2361
WOODW	1098×8418	1.3044
WOOD1P	244×2595	1.4429

5.3.2 Aspectos de Implementação

Na implementação das inicializações avançadas, três parâmetros adicionais foram considerados:

k– Determina o número de iterações que serão realizadas pelas redes de Hopfield modificada com as direções afins escala; foi considerado $k = 1$ no total dos problemas testados;

T– Tolerância de convergência do processo de factibilização; foi considerado $T = 10^{-5}$ na maioria dos casos;

It– Parâmetro de interioridade, utilizado na função de ativação; valor ajustado em função do problema.

5.3.3 Avaliação dos Resultados Computacionais com o Método Primal

A Tabela 5.2 apresenta os resultados obtidos pelo método primal afim escala utilizando as três alternativas de inicialização. Para cada inicialização, são mostrados: o número de iterações do método primal afim e o tempo total de processamento. Esta tabela está dividida em quatro partes, descritas a seguir.

1. PROBLEMA $(m \times n)$. Identifica o problema testado, assim como a sua dimensão.

2. ABORDAGEM TRADICIONAL. Representa os resultados obtidos pelo método primal afim escala a partir de um ponto inicial calculado pela inicialização tradicional FASE I. Está dividida em três colunas. A primeira coluna mostra o número de iterações da FASE II. A segunda coluna apresenta o número total de iterações obtido pela soma das iterações realizadas no cálculo do ponto inicial por FASE I, mais as iterações realizadas pelo método primal afim escala até a convergência (FASE II). A terceira coluna ilustra o tempo total de processamento (incluindo as duas fases).

3. FACTIBILIZAÇÃO-PRIMAL. Representa os resultados obtidos pelo método primal afim escala a partir do ponto inicial calculado pelo processo de factibilização. Está dividida em duas colunas, com o número de iterações e o tempo de processamento. As iterações consideradas são as realizadas pelo método primal afim escala a partir do ponto inicial. O tempo total considera os tempos gastos pelo processo de

factibilização mais o tempo para a otimização do problema pelo método primal afim escala.

4. HOPFIELD-PRIMAL. Representa os resultados obtidos pelo método primal afim escala a partir do ponto inicial avançado calculado pela abordagem Hopfield-primal. Está dividida em duas colunas, com o número de iterações e o tempo de processamento. Considera as iterações realizadas pelo método Hopfiel-primal ($k = 1$) na inicialização, mais as realizadas pelo método primal na convergência. O tempo de processamento é medido da mesma forma que nos casos anteriores, considerando as duas etapas do processo de otimização.

TABELA 5.2. RESULTADOS OBTIDOS COM O MÉTODO PRIMAL AFIM ESCALA.

PROBLEMA $(m \times n)$		ABORDAGEM TRADICIONAL			FACTIBILIZAÇÃO-PRIMAL		HOPFIELD-PRIMAL	
		FASE II ITER	TOTAL ITER	TEMPO (S)	ITER	TEMPO (S)	ITER	TEMPO (S)
AFIRO	(27x51)	21	29	0.34	13	2.03	11	13.86
SC50B	(48x76)	23	27	0.34	16	8.20	13	6.61
SC50A	(49x77)	30	34	0.39	16	7.34	15	2.62
SC105	(104x162)	52	57	0.76	20	40.92	15	8.87
ADLITTLE	(55x137)	49	54	0.53	43	13.19	60	56.97
SCAGR7	(128x184)	101	113	1.47	31	5.48	49	78.56
STOCFOR1	(109x157)	45	62	0.98	48	29.11	33	44.25
BLEND	(74x114)	51	57	0.80	27	125.37	18	80.78
SC205	(203x315)	109	115	1.91	22	47.47	19	36.01
SHARE2B	(96x112)	62	85	1.31	34	59.50	29	162.94
SCTAP1	(300x660)	29	43	3.61	27	58.98	27	129.54
BRANDY	(149x259)	-	-	-	74	43.48	41	604.11
SCSD1	(77x760)	22	28	1.16	17	1.52	15	3.23
BANDM	(269x436)	182	207	10.69	118	180.74	51	187.82
BEACONFD	(148x270)	121	132	5.67	38	2.30	30	78.84
SCSD6	(147x1350)	20	26	2.78	13	2.16	13	4.08
SCSD8	(397x2750)	18	23	10.16	17	7.75	17	12.86
TRUSS	(1000x8806)	31	36	107.50	25	70.92	31	91.50
WOODW	(1098x8418)	90	110	181.65	75	109.14	83	115.70
WOOD1P	(244x2595)	80	96	40.98	-	-	-	-

(**–**) não foi obtida a convergência do método.

Observando-se a Tabela 5.2, pode-se estabelecer as considerações a seguir.

1. Em todos os casos estudados, a cooperação de redes neurais de Hopfield com o método primal afim escala permitiu reduzir o número de iterações na FASE II do processo de otimização (por melhores razões, o número total de iterações foi também reduzido).

2. Observa-se que o número de iterações do método primal afim escala tem forte relação com o ponto inicial. Em quase todos os casos apresentados, o método beneficia-se da inicialização avançada por Hopfield. Por exemplo, pela FASE I foi obtido um ponto inicial para o problema BRANDY em *38* iterações, mas o método afim escala não convergiu a partir deste ponto; no entanto, com os pontos iniciais calculados pelas abordagens com redes de Hopfield, foi obtida a convergência do método primal afim escala ao ótimo.

3. Na maioria dos casos estudados, o tempo total para obtenção de soluções ótimas foi maior nas situações de cooperação com as redes de Hopfield do que com a utilização da metodologia tradicional para inicialização de métodos de pontos interiores.

Uma avaliação das várias etapas das alternativas de cooperação, entre redes de Hopfield e métodos de pontos interiores, mostrou que a maior parte do tempo computacional é usada na fase de factibilização. De fato, o estudo de abordagens alternativas para a factibilização das redes de Hopfield pode ser um desdobramento atraente deste trabalho.

Observou-se que pontos "mais interiores" tendem a reduzir o número de iterações da FASE II do método primal afim escala. Em principio, é possível obter esses pontos, aumentando-se os parâmetros *It* das funções de ativação, utilizadas na fase de factibilização. No entanto, observou-se que esse aumento de interioridade dificulta a convergência da fase de factibilização.

A influência que tem a interioridade do ponto inicial na convergência do método primal afim escala pode ser exemplificada com o problema WOOD1P. A fase de factibilização encontrou um ponto inicial interior factível, mas muito perto da fronteira (isto é, "pouco interior"): o método primal afim escala não conseguiu convergir a partir deste ponto.

O indicador de que a cooperação de redes de Hopfield com o método primal afim tende a se tornar mais atraente nos problemas grandes deve ser visto com certa cautela no estágio atual da pesquisa. Nos maiores problemas testados, as alternativas de cooperação tiveram os tempos computacionais menores do que os obtidos na abordagem tradicional do método primal afim escala; no entanto, a obtenção de pontos iniciais para estes problemas é relativamente simples, reduzindo o esforço da fase de factibilização.

5.3.4 Avaliação dos Resultados Computacionais com o Método Dual Afim

A Tabela 5.3 mostra os resultados obtidos pelo método dual afim escala, com as três inicializações consideradas. Para cada inicialização são mostrados o número de iterações do método dual afim escala e o tempo total de processamento. Esta tabela está dividida em quatro partes, detalhadas a seguir.

1. PROBLEMA $(m \times n)$. Esta coluna identifica o problema testado, assim como a sua dimensão.

2. ABORDAGEM TRADICIONAL. Mostra os resultados obtidos pelo método dual afim escala a partir do ponto inicial calculado pela inicialização tradicional por FASE I. Os resultados são apresentados em três colunas. As duas primeiras mostram o número de iterações da FASE II na otimização e o número total de iterações, obtido pela soma das iterações da FASE I (cálculo do ponto inicial) mais as iterações da FASE II (método dual afim escala até a convergência). O tempo total de processamento, incluindo as duas fases, é mostrado na terceira coluna.

3. FACTIBILIZAÇÃO-DUAL. Mostra os resultados obtidos pelo método dual afim escala a partir do ponto inicial calculado pelo processo de factibilização para o dual. Está dividida em duas colunas, com o número total de iterações e o tempo total de processamento das duas etapas. As iterações consideradas são as utilizadas pelo método dual na convergência a partir do ponto inicial.

4. HOPFIELD-DUAL. Apresenta os resultados obtidos pelo método dual afim escala a partir do ponto inicial avançado calculado pela abordagem Hopfield-Dual. Os resultados são mostrados em duas colunas, com o número total de iterações e tempo total de processamento das duas etapas de otimização. O número total de iterações inclui as iterações do método Hopfield-dual ($k = 1$) na inicialização e as realizadas pelo método dual afim escala, até a convergência.

TABELA 5.3. RESULTADOS OBTIDOS COM O MÉTODO DUAL AFIM ESCALA.

PROBLEMA $(m \times n)$	ABORDAGEM TRADICIONAL			FACTIBILIZAÇÃO-DUAL		HOPFIELD-DUAL	
	FASE II ITER	TOTAL ITER	TEMPO (S)	ITER	TEMPO (S)	ITER	TEMPO (S)
AFIRO (27x51)	19	20	0.20	14	0.23	15	0.24
SC50B (48x76)	19	21	0.20	14	0.31	14	0.30
SC50A (49x77)	18	19	0.20	15	0.36	16	0.40
SC105 (104x162)	21	24	0.30	16	1.31	17	1.17
ADLITTLE (55x137)	22	23	0.25	19	9.50	23	19.81
SCAGR7 (128x184)	39	43	0.47	23	2.66	25	2.70
BLEND (74x114)	20	25	0.31	20	1.72	21	4.43
SC205 (203x315)	23	28	0.44	24	338.95	31	509.41
SHARE2B (96x162)	22	26	0.34	19	88.76	23	255.92
SHARE1B (112x248)	35	40	0.61	-	-	-	-
SCAGR25 (470x670)	25	28	1.25	50	18.17	29	35.15
SCTAP1 (300x660)	27	33	1.31	87	4.84	51	18.03
SCSD1 (77x760)	16	19	0.70	16	0.61	17	0.73
BANDM (269x436)	23	31	0. 90	25	508.27	25	878.48
E226 (220x469)	-	-	-	51	469.05	58	998.87
BEACONFD (148x270)	-	-	-	15	5.61	34	77.34
SCSD6 (147x1350)	18	21	2.06	19	1.89	19	16.11
SCFXM2 (644x1184)	-	-	-	48	1362.54	45	4057.59
SCSD8 (397x2750)	19	23	8.23	19	6.81	16	151.17
SCTAP3 (1480x3340)	27	35	37.64	30	57.70	29	79.86
SHIP121 (838x5329)	22	28	29.97	49	112.85	84	213.70
TRUSS (1000x8806)	28	36	4.09	29	5.66	-	-
WOODW (1098x8418)	55	56	51.36	41	393.70	42	397.45
WOOD1P (244x2595)	39	40	15.22	22	22.05	24	28.81

(**–**) não foi obtida a convergência do método.

A análise dos resultados resumidos na Tabela 5.3, obtidos pelo método dual afim escala com as três alternativas de inicialização, permite estabelecer as seguintes considerações.

1. Na grande maioria dos casos estudados, a cooperação das técnicas de redes de Hopfield e do método dual afim escala permitiu reduzir o número total de iterações do processo de otimização. Em muitos dos problemas foi possível diminuir o número de iterações da FASE II. No entanto, para alguns problemas não foi possível diminuir as

iterações do método dual afim escala com o ponto inicial obtido pelas abordagens com redes de Hopfield (problemas SCAGR25, SCTAP1 e SHIP12L).

2. Da mesma forma que para o método primal afim escala, pode-se observar que o número de iterações do método dual afim tem forte relação com o ponto inicial. Por exemplo, para os problemas BEACONFD, E226 e SCFXM2 a inicialização tradicional por FASE I não encontrou um ponto inicial factível, mas as abordagens por redes de Hopfield encontraram um ponto interior factível e foi possível obter a convergência do método, a partir do ponto obtido.

3. Na maioria dos problemas estudados, o tempo total na otimização, incluindo as duas etapas de processamento, foi maior nas situações em que eram aplicadas as abordagens cooperativas do que com a utilização da inicialização tradicionalmente utilizada para este método.

Avaliando-se as várias etapas das metodologias cooperativas entre redes de Hopfield e o método dual afim escala, foi observado que a maior parte do tempo total de processamento é usada na fase de factibilização para o cálculo de um ponto interior factível.

Nos testes realizados com diferentes valores para o parâmetro interioridade (*It*), para um mesmo problema, pudemos identificar um comportamento geral: o aumento do valor do parâmetro *It*, tornando os pontos "mais interiores", permite diminuir o número de iterações do método dual afim escala na otimização. Este comportamento foi também observado no estudo de casos com o método primal afim escala. De forma complementar, quando o ponto é muito perto da fronteira (menos interior) o método pode não convergir, como ocorreu com os pontos obtidos pelas abordagens cooperativas para o problema SHARE1B.

CAPÍTULO 6.

OTIMIZAÇÃO POR REDES NEURAIS E MÉTODOS PRIMAIS-DUAIS DE PONTOS INTERIORES

Possibilidades de cooperação entre as técnicas de redes de Hopfield modificadas e métodos de pontos interiores foram exploradas no capítulo anterior através das inicializações desenvolvidas para os métodos afins. Este capítulo continua esta linha de investigação, avaliando possibilidades de cooperação com a família de métodos primais-duais de pontos interiores.

Dois conjuntos de possibilidades de cooperação de redes de Hopfield com a família de métodos primais-duais são avaliados, considerando alternativas com o método de pontos interiores primal-dual clássico e com o método preditor-corretor. O estudo de casos considera problemas da biblioteca *Netlib*.

6.1 INICIALIZAÇÃO TRADICIONAL PARA OS MÉTODOS PRIMAIS-DUAIS

O método de pontos interiores primal-dual e o método preditor-corretor utilizam simultaneamente um ponto interior primal (x) e um ponto interior dual (y,z); é usual utilizar-se a denominação "ponto primal-dual" (x,y,z) para se referir simultaneamente aos dois pontos.

Como foi discutido no Capítulo 3, o ponto interior primal-dual, para inicialização do processo de otimização por métodos primais-duais de pontos interiores, não precisa necessariamente ser factível. Esse aspecto facilita a obtenção de pontos iniciais para os métodos de pontos interiores primais-duais. No entanto, deve-se ressaltar que o estudo de casos apresentado neste capítulo mostra que a convergência dos métodos primais-duais pode ser melhorada se os pontos iniciais forem factíveis.

A seguir, apresenta-se uma metodologia analítica, proposta por Lustig e co-autores [1992] para obtenção de pontos interiores inicias primais-duais (não necessariamente factíveis); ou seja a metodologia faz o papel de FASE I nos métodos primais-duais. Para simplificar a exposição, apresenta-se separadamente a obtenção de ponto interior primal e de ponto interior dual.

Ponto Interior Primal. As componentes do ponto *x* são calculadas pela Equação (6.1),

$$x_i = max(\tilde{x}_i, E_1), \tag{6.1}$$

onde:

$\tilde{x} = A^T \cdot \left(A \cdot A^T\right)^{-1} \cdot b$;

$E_1 = max\left(-min(\tilde{x}_i), E_2, \frac{\|b\|}{E_2 \cdot \|A\|}\right)$;

$\|A\| = \max_j \sum_j |a_{ij}|$;

$\|b\| = \sum_i |b_i|$ e

$E_2 = 100$ (na maioria das implementações).

Ponto Interior Dual. O ponto inicial (y, z) é calculado pelas Equações (6.2) e (6.3),

$$y = 0, \tag{6.2}$$

$$z = \begin{cases} c_i + E_3 & c_i \geq 0 \\ -c_i & c_i \leq -E_3 \\ E_3 & -E_3 \leq c_i \leq 0 \end{cases}, \tag{6.3}$$

onde $E_3 = 1 + \|c\|$ e $\|c\| = \sum_i |c_i|$.

A metodologia procura obter um ponto inicial bem posicionado (ou seja, "mais interior") para melhorar a convergência dos métodos primais-duais. Vale lembrar que "bem posicionado" significa um ponto não muito próximo da fronteira.

6.2 Inicialização por Cooperação com Redes de Hopfield Modificadas

Propõem-se duas abordagens para inicialização avançada dos métodos primais-duais por cooperação entre redes de Hopfield e métodos primais-duais de pontos interiores; ambas utilizam os aperfeiçoamentos acrescentados às redes de Hopfield modificadas [Silva 1997], discutidos no Capítulo 4 e Capítulo 5. Denominaremos as abordagens de Hopfield-primal-dual e Hopfield-preditor-corretor, para especificar, respectivamente a cooperação com o método primal-dual de pontos interiores e a cooperação com o método preditor-corretor.

As abordagens Hopfield-primal-dual e Hopfield-preditor-corretor diferem apenas em relação à direção $(\Delta x^k, \Delta y^k, \Delta z^k)$ utilizada no Passo II do processo para obtenção de pontos interiores avançados, descrito a seguir. A direção $(\Delta x^k, \Delta y^k, \Delta z^k)$ para a

abordagem Hopfield-primal-dual é a mesma direção adotada no método de pontos interiores primal-dual, discutido no Capítulo 3. De forma análoga, a direção $\left(\Delta x^k, \Delta y^k, \Delta z^k\right)$ para a abordagem Hopfield-preditor-corretor foi herdada do método preditor-corretor apresentado no Capítulo 3.

Para ambas as abordagens, um ponto interior primal-dual avançado pode ser obtido com a seqüência de passos descrita a seguir.

I. Calcular um ponto inicial primal-dual pelo processo de factibilização.

   i. Calcular um ponto primal interior pelo processo de factibilização para o primal.

   ii. Calcular um ponto dual interior pelo processo de factibilização para o dual.

II. Calcular o próximo ponto primal-dual

   i. Calcular a direção $\left(\Delta x^k, \Delta y^k, \Delta z^k\right)$;

   ii. Calcular o próximo ponto, $\left(x^{k+1}, y^{k+1}, z^{k+1}\right) = \left(x^k, y^k, z^k\right) + \alpha \cdot \left(\Delta x^k, \Delta y^k, \Delta z^k\right)$

III. Calcular um ponto primal-dual factível pelo processo de factibilização.

   i. Calcular um ponto primal factível a partir do ponto x^{k+1} pelo processo de factibilização.

   ii. Calcular um ponto dual factível a partir do ponto $\left(y^{k+1}, z^{k+1}\right)$ pelo processo de factibilização para o dual.

IV. Se o ponto primal-dual $\left(x^{k+1}, y^{k+1}, z^{k+1}\right)$ for considerado suficientemente avançado, utilize o método primal-dual para conclusão da otimização; caso contrário, incremente o contador *k* e volte ao Passo II.

Como já discutido no capítulo anterior, uma alternativa simples para se obter um ponto “suficientemente avançado” é através de adoção de um número fixo de iterações.

6.3 CÁLCULO DO PASSO

É possível argumentar que o valor unitário para o passo fixo α, utilizado no processo de atualização da inicialização avançada, é a melhor escolha. O argumento pode ser sintetizado em três pontos, enumerados a seguir.

1. Quando $\alpha = 1$, $\left(x^{k+1}, y^{k+1}, z^{k+1}\right)$ é factível em relação às restrições de igualdade, isto é, $A \cdot x = b,\ A^T \cdot y + z = c$. Por outro lado, as direções $\left(\Delta x^k, \Delta y^k, \Delta z^k\right)$ são baseadas no método de Newton [Luenberger 1984] e o melhor passo em uma direção de Newton é $\alpha = 1$ (o que significa aproveitamento de *100%* da direção para um problema linear).

2. Quando $\alpha < 1$, $\left(x^{k+1}, y^{k+1}, z^{k+1}\right)$ não é factível e não são exploradas todas as possibilidades da direção (em relação à otimalidade).

3. Quando $\alpha > 1$, $\left(x^{k+1}, y^{k+1}, z^{k+1}\right)$ não é factível.

6.4 ESTUDO DE CASOS

Esta seção apresenta o estudo de casos realizado com os métodos primal-dual clássico e preditor-corretor, utilizando as cooperações propostas com as redes de Hopfield modificadas. Nesse estudo de casos, foram também utilizados problemas da biblioteca *Netlib*. Observa-se que se utilizou um número maior de problemas do que no Capítulo 5.

O estudo de casos foi realizado em um computador PC, Pentium IV, 2.26 GHz, 512 Mb de memória com sistema operacional Windows 2000.

O estudo é apresentado separadamente para cada método. O desempenho do método primal-dual clássico é avaliado com duas inicializações: inicialização tradicional, discutida na Seção 6.1, inicialização avançada por cooperação com redes de Hopfield. Na comparação, são considerados o número de iterações e o tempo total de processamento.

De forma análoga, o desempenho do método preditor-corretor é avaliado com a inicialização tradicional, discutida na Seção 6.1, e com a inicialização avançada, que caracteriza a abordagem Hopfield-preditor-corretor. Para essa avaliação, foi possível utilizar um código de referência na área denominado *LIPSOL* [Zhang 1998] mencionado no Capítulo 3. Este código, programado em *Matlab* e *Fortran*, é disponível através da Internet (http://www.caam.rice.edu/~zhang/lipsol/).

6.4.1 Aspectos de Implementação

Cinco parâmetros foram utilizados para a aplicação das abordagens cooperativas:

k- Determina o número de iterações realizadas pelas redes de Hopfield modificadas com as direções primais-duais; foi utilizado $k = 1$ na maioria dos problemas testados;

i- Número de iterações do processo de factibilização; adotou-se $i = 100$;

It_p- Interioridade do ponto primal utilizada na função de ativação; valor ajustado em função do problema;

It_d- Interioridade do ponto dual utilizada na função de ativação para o dual; valor ajustado em função do problema.

6.4.2 Avaliação dos Resultados Computacionais com o Método Primal-Dual

A Tabela 6.1 apresenta os resultados obtidos pelo método primal-dual clássico utilizando as duas inicializações: inicialização tradicional (discutida na Seção 6.1) e inicialização avançada com a abordagem Hopfield-primal-dual. O desempenho do método primal-dual foi avaliado em relação ao número de iterações até a convergência e ao tempo total de processamento. A Tabela 6.1 está dividida em três partes principais, descritas a seguir:

1. PROBLEMA $(m \times n)$. Identifica o problema testado, assim como a sua dimensão.

2. ABORDAGEM TRADICIONAL. Representa os resultados obtidos pelo método primal-dual clássico utilizando o ponto inicial tradicionalmente utilizado para este método (Seção 6.1). Os resultados são divididos em duas colunas; número de iterações do método primal-dual clássico e o tempo de processamento até a convergência.

3. HOPFIELD-PRIMAL-DUAL. Representa os resultados obtidos pelo método primal-dual clássico utilizando o ponto inicial avançado. O número de iterações inclui as iterações da inicialização avançada ($k = 1$), mais as iterações do método primal-dual na otimização. O tempo apresentado considera as duas etapas do processo de otimização.

Avaliando-se os resultados resumidos na Tabela 6.1 pode-se estabelecer as considerações a seguir.

1. Em todos os casos estudados, o método primal-dual clássico com a inicialização avançada alcançou a otimalidade com menor número de iterações do que o método primal-dual clássico com a inicialização tradicional.

2. Em todos os estudos de casos apresentados, o tempo total de processamento do método primal-dual com inicialização pela abordagem Hopfield-primal-dual foi maior do que quando era usada a inicialização tradicional.

No estudo de casos, observou-se também que o melhor desempenho do método era obtido quando o ponto era bem interior à região de factibilidade. Este bom posicionamento (ponto mais centrado) é obtido quando aumentamos os valores dos parâmetros de interioridade (It_p, It_d) para o ponto primal-dual nas funções de ativação.

O passo mais caro em termos de tempo de processamento, no processo de inicialização avançada, é a fase de factibilização. Por outro lado, as vantagens da factibilização e interioridade (por aumento dos parâmetros It_p, It_d da função de ativação) foram as causas principais para diminuição do número de iterações do método primal-dual.

TABELA 6.1. RESULTADOS OBTIDOS PELO MÉTODO PRIMAL-DUAL CLÁSSICO.

PROBLEMA $(m \times n)$		**ABORDAGEM TRADICIONAL**		**HOPFIELD-PRIMAL-DUAL**	
		ITER	**TEMPO(S)**	**ITER**	**TEMPO(S)**
AFIRO	(27x51)	11	0.08	8	0.31
SC50B	(48x76)	10	0.08	7	0.37
SC50A	(49x77)	12	0.08	8	0.37
SC105	(104x162)	14	0.09	13	2.56
ADLITTLE	(55x137)	19	0.11	11	1.40
SCAGR7	(128x184)	19	0.12	17	1.37
STOCFOR1	(109x157)	21	0.14	16	3.20
BLEND	(74x114)	18	0.13	12	0.56
SC205	(203x315)	14	0.11	14	1.00
SHARE2B	(96x162)	19	0.13	14	0.79
LOTFI	(151x364)	29	0.24	27	1.28
SHARE1B	(112x248)	32	0.22	29	0.89
SCAGR25	(470x670)	27	0.29	21	1.77
SCTAP1	(300x660)	34	0.39	24	2.23
BRANDY	(149x259)	24	0.29	24	1.61
ISRAEL	(174x316)	33	0.80	27	1.92
SCSD1	(77x760)	12	0.12	8	1.68
AGG	(488x615)	28	1.34	26	2.51
BANDM	(269x436)	32	0.89	21	3.63
E226	(220x469)	37	0.56	28	2.91
BEACONFD	(148x270)	19	0.26	11	1.55
SCSD6	(147x1350)	15	0.19	11	2.79
SCFXM2	(644x1184)	41	1.16	34	5.14
FFFFF800	(501x1005)	39	5.65	37	13.94
SHIP12S	(466x2293)	35	2.57	24	4.61
STOCFOR2	(2157x3054)	50	56.09	39	55.47
SCSD8	(397x2750)	17	0.33	12	8.79
SCTAP3	(1480x3340)	31	5.94	18	29.81
25FV47	(798x1854)	46	14.84	44	82.15
SHIP121	(838x5329)	38	28.91	29	50.80
TRUSS	(1000x8806)	25	15.71	18	50.39
D2Q06C	(2171x5831)	67	126.18	63	144.14
WOODW	(1098x8418)	54	14.19	32	19.11

Em alguns problemas testados, foram obtidos melhores resultados em número de iterações quando era diminuída a tolerância de convergência do processo de factibilização (T_p e T_d). Por exemplo, com os problemas AFIRO e STOCFOR1 foi possível diminuir o número de iterações quando foram usados os valores de tolerância $T_p = 10^{-2}$ e $T_d = 10^{-2}$, mas o tempo total de processamento foi comprometido como mostra a Tabela 6.2.

TABELA 6.2. RESULTADOS OBTIDOS PELO MÉTODO PRIMAL-DUAL CLÁSSICO COM MENOR TOLERÂNCIA DE CONVERGÊNCIA NA PROJEÇÃO.

PROBLEMA	ITER	TEMPO(S)
AFIRO	7	2.088
STOCFOR1	11	125.5543

6.4.3 Avaliação dos Resultados Computacionais com o Método Preditor-Corretor

O desempenho do método preditor-corretor com inicialização tradicional é comparado com o seu desempenho quando utilizada a inicialização avançada da abordagem Hopfield-preditor-corretor. Como já foi mencionado, utilizou-se a implementação *LIPSOL* do método preditor-corretor, o qual adota a inicialização tradicional, discutida na Seção 6.1.

As abordagens são comparadas apenas em relação ao número de iterações. Embora fosse desejável termos também algum indicador de tempo total de processamento, esses dados só seriam informativos se os métodos fossem codificados na mesma linguagem e utilizassem recursos semelhantes de álgebra matricial. Infelizmente, isto não foi possível; o Hopfield-preditor-corretor está codificado em *Matlab*, enquanto o *LIPSOL* possui rotinas implementadas em linguagem *Fortran*, com aspectos de álgebra matricial otimizados.

Na Tabela 6.3, são resumidos os resultados obtidos pelo preditor-corretor com as duas inicializações consideradas.

O método preditor-corretor apresenta melhor desempenho entre os métodos anteriormente estudados. É um método muito estável e a sua convergência é sempre garantida independentemente do ponto inicial. Nota-se que, em todas as instâncias de problemas, o método Hopfield-preditor-corrector alcançou a solução ótima em menor número de iterações do que o *LIPSOL*. O tempo total de processamento foi sempre menor para o *LIPSOL*, mas, como já observado, esta informação não permite qualquer inferência significativa.

Comparando-se a Tabela 6.3 com as Tabelas 5.2, 5.3 e 6.1, observou-se que em todos os problemas o número de iterações do método preditor-corretor até a convergência é inferior, comparado com o número de iterações dos outros métodos estudados, para a solução dos mesmos problemas. No entanto, uma iteração do método preditor-corretor realiza um número maior de operações do que os demais métodos de pontos interiores.

TABELA 6.3. RESULTADOS OBTIDOS PELO PREDITOR-CORRETOR COM INICIALIZAÇÃO POR HOPFIELD-PREDITOR-CORRETOR E PELO PREDITOR-CORRETOR DO *LIPSOL*.

PROBLEMA $(m \times n)$		**LIPSOL**	**HOPFIELD-PREDITOR-CORRETOR**
AFIRO	(27x51)	8	6
SC50B	(48x76)	7	5
SC50A	(49x77)	10	6
SC105	(104x162)	10	9
ADLITTLE	(55x137)	13	9
SCAGR7	(128x184)	14	11
STOCFOR1	(109x157)	16	12
BLEND	(74x114)	12	8
SC205	(203x315)	10	9
SHARE2B	(96x162)	13	10
LOTFI	(151x364)	18	14
SHARE1B	(112x248)	22	19
SCAGR25	(470x670)	17	12
SCTAP1	(300x660)	17	12
BRANDY	(149x259)	17	15
ISRAEL	(174x316)	23	15
SCSD1	(77x760)	9	6
AGG	(488x615)	21	16
BANDM	(269x436)	18	14
E226	(220x469)	21	14
BEACONFD	(148x270)	13	6
SCSD6	(147x1350)	11	8
SCFXM2	(644x1184)	21	20
FFFFF800	(501x1005)	26	23
SHIP12S	(466x2293)	18	14
STOCFOR2	(2157x3054)	21	20
SCSD8	(397x2750)	11	7
SCTAP3	(1480x3340)	18	16
25FV47	(798x1854)	25	20
SHIP121	(838x5329)	18	16
TRUSS	(1000x8806)	19	17
D2Q06C	(2171x5831)	32	28
WOODW	(1098x8418)	28	26

CONCLUSÕES

O trabalho explora alternativas de cooperação entre redes neurais e métodos de pontos interiores para a solução de problemas de otimização linear. Essencialmente, utilizou-se redes de Hopfield com as modificações propostas por Silva [1997] para produzir pontos interiores iniciais ou pontos interiores avançados na direção de otimalidade; esses pontos são entregues a métodos de pontos interiores, que concluem o processo de otimização.

Investigaram-se alternativas de cooperação de redes de Hopfield com as principais famílias de métodos de pontos interiores para programação linear: métodos afins e métodos primais-duais. Ao todo, foram desenvolvidas alternativas de cooperação com quatro métodos: primal afim, dual afim, primal-dual clássico e preditor-corretor.

As principais contribuições do trabalho podem ser resumidas nos pontos a seguir.

1. Aperfeiçoamentos nas redes de Hopfield modificadas, incluindo a especialização de resultados recentes de álgebra matricial e o uso de direções mais ricas, herdadas dos métodos de pontos interiores. Esses aperfeiçoamentos permitiram abordar, através de redes neurais, os problemas reais de otimização linear da biblioteca *Netlib* (apresentados no Capítulo 3). A avaliação da bibliografia na área indica que

problemas de otimização com essas dimensões até então não haviam sido abordados por técnicas de redes neurais.

2. Desenvolvimento de duas alternativas para cooperação de redes de Hopfield com o método primal afim escala de pontos interiores. A primeira alternativa utiliza uma versão aperfeiçoada do processo de factibilização das redes de Hopfield para produzir um ponto inicial interior factível para o método primal afim; este procedimento substitui a abordagem tradicional, denominada "FASE I", para obtenção de pontos interiores iniciais. A segunda alternativa avança na direção da otimalidade com as redes de Hopfield, proporcionando inicialização "a quente" para o método primal afim.

3. Elaboração de duas alternativas de cooperação de redes de Hopfield com o método dual afim escala, seguindo linhas semelhantes às desenvolvidas para o método primal afim.

4. Desenvolvimento de alternativa de inicialização avançada por redes de Hopfield para o método primal-dual clássico de pontos interiores.

5. Construção de procedimento para inicialização "a quente" para o método preditor-corretor de pontos interiores.

6. Estudo de um amplo conjunto de casos para avaliação de todas as abordagens desenvolvidas.

A redução da trajetória para obtenção de soluções ótimas foi o aspecto mais bem sucedido do conjunto de alternativas propostas neste trabalho, envolvendo a cooperação entre redes de Hopfield e métodos de pontos interiores. Esta característica foi quantificada na redução do número de iterações dos métodos de pontos interiores, observada na ampla maioria dos casos estudados.

Por outro lado, os tempos de processamento para obtenção de soluções ótimas não acompanharam a redução das trajetórias; quase sempre, foram maiores nas abordagens

cooperativas do que com as abordagens tradicionais para inicialização dos métodos de pontos interiores.

O aparente descompasso entre as duas medidas de desempenho (número de iterações e tempo de processamento) pode ser compreendido quando examinamos detalhadamente as várias etapas do processo de otimização. Essa avaliação revelou que uma parte substantiva do tempo de processamento é usado na fase de factibilização das redes de Hopfield modificadas. Paradoxalmente, esta etapa envolveu operações simples; no entanto, as operações são exaustivamente repetidas para produzir pontos interiores factíveis. O aperfeiçoamento da etapa de factibilização das redes de Hopfield modificadas é um desdobramento promissor das investigações realizadas neste trabalho.

Deve-se também ressaltar que alguns aspectos da cooperação entre redes de Hopfield e métodos de pontos interiores foram explorado apenas de forma preliminar. Entre esses, destaca-se o uso de uma única fatoração simbólica para utilização tanto nas redes de Hopfield como nos métodos de pontos interiores. Para isso, é necessário o desenvolvimento de códigos em linguagem que permita estruturação detalhada de todas as operações matriciais.

Para concluir, deve-se ressaltar que as investigações realizadas neste trabalho mostram que existem aspectos promissores e ainda pouco explorados na fronteira entre as áreas de redes neurais e métodos de otimização. A sondagem dessa fronteira poderia trazer benefícios para as duas áreas.

BIBLIOGRAFIA

Adler, I., M. G. C. Resende, G. Veiga and N. Karmarkar (1989). An Implementation of Karmarkar's Algorithm for Linear Programming. *Mathematical Programming*, vol. 44, pp. 297—335.

Aiyer, S. V. B., M. Niranja and F. Fallside (1990). A Theoretical Investigation into the Performance of the Hopfield Model. *IEEE Trans. Neural Networks*, vol. 1, nº 2, pp. 204—215.

Aiyer S. V. B. and F. Fallside (1991). A Subspace Approach to Solving Combinatorial Optimization Problems with Hopfield Networks. CUED/F-INFENG/TR 55, URL: http://mi.eng.cam.ac.uk/reports/abstracts/aiyer_tr55.html, University of Cambridge, England.

Barbosa, V. C. and L. A. V de Carvalho (1990). Feasible Directions Linear Programming by Neural Networks. *Proceeding of The IJNN-International Joint Conference on Neural Networks*, vol. 3, pp. 941—946.

Bazaraa, M. S., J. J. Jarvis and H. D. Sherali (1997). *Linear Programming and Network Flows*. Second Edition, John Wiley & Sons.

Bixby, R. E. (1992). Implementing The Simplex Method: The Initial Basis. *ORSA Journal on Computing*, vol. 4, nº 3, pp. 267-284.

Bouzerdoum, A. and T. R. Pattison (1993). Neural Network for Quadratic Optimization with Bound Constraints. *IEEE Trans. Neural Networks,* vol. 4, nº 2, pp. 293—304.

Castro, L. N De, E. M Iyoda, F. J. Von Zuben and R. R. Gudwin (1998). Feedforward Neural Network Initialization: an Evolutionary Approach. *Proceedings of the IEEE SBRN'98 (Brazilian Symposium on Neural Networks),* pp. 43—48, Belo Horizonte/MG, Brazil.

Chellapilla K. and D. B. Fogel (1999). Evolution, Neural Networks, Games and Intelligence. *Proceedings of the IEEE,* vol. 87, nº 9, pp. 1471—1496.

Chellapilla K. and D. B. Fogel (1999a). Evolving neural networks to play checkers without relying on expert knowledge. *IEEE Trans. on Neural Networks,* vol. 10, nº 6, pp. 1382—1391.

Chellapilla K. and D. B. Fogel, (2000). "Anaconda Defeats Hoyle 6-0: A case Study Competing an evolved checkers program against commercially available software" *Proc. of the 2000 Congress on Evolutionary Computation*, pp. 857—863.

Chiu, C., C.-Y. Maa and M. A. Shanblatt (1991). Energy Function Analysis of Dynamic Programming Neural Networks. *IEEE Trans. on Neural Networks*, vol. 2, nº 4, pp. 418—426.

Chua, L. O. (1990). Global Optimization: A naive Approach. *IEEE Trans. on Circuits and Systems*, vol.37, nº 7, pp. 966—969.

Dikin, I. I. (1967). Iterative solution of problems of linear and Quadratic programming. *Soviet Mathematics Doklaly*, vol. 8, pp. 674—675.

Duff, S. I., A. M. Erisman and J. K. Reid (1986). *Direct Methods for Sparse Matrices.* Clarendon Press, Oxford.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, vol. 2, nº 3, pp. 183—192.

Gay, D. M. (1985). Electronic Mail Distribution of Linear Programming Test Problems. *Mathematical Programming Society COAL Newsletter*, vol. 13, (Dec.), pp.10—12.

Golub, G. H. and C. F. Van Loan (1996). *Matrix Computations.* Third Edition, The Johns Hopkins University Press, Baltimore.

Gondzio, J. (1996). Warm-Start of the Primal-Dual Method Applied in the Cutting-Plane Scheme. Logilab TR 96.3, URL: http://www.unige.ch/hec/logilab/templeet/templeet.php/rapport.fr.html, University of Geneva, Switzerland.

Gondzio, J. and J.-P. Vial (1997). Warm-Start and ε-subgradients in cutting plane schema for block-angular linear programs. Logilab TR 1997.1, URL: http://www.unige.ch/hec/logilab/templeet/templeet.php/rapport.fr.html, University of Geneva, Switzerland.

Gondzio, J. (1997a). Presolve Analysis of Linear Programs Prior to Applying an Interior point Method. *INFORMS Journal on Computing*, vol. 9, nº 1, pp. 73—91.

Harrald, P.G. and M. Kamstra (1997). Evolving artificial neural networks to combine financial forecasts. *IEEE Trans. on Evolutionary Computation,* vol. 1, nº 1, pp. 40—52.

Haykin, S. U. (1999). *Neural Networks: A Comprehensive Foundation*. Second Edition, Prentice Hall Inc., Upper Saddle River, New Jersey, USA.

Herrera, F., M. Lozano y J. L. Verdegay (1994). Algoritmos Genéticos: Fundamentos, Extensiones y Aplicaciones. DECSAI TR #DECSAI 94105, URL: http://decsai.ugr.es/~herrera/public1.html#IntJou, Universidad de Granada, España.

Hertz, J., A. Krogh and R. G. Palmer (1991). Introduction to The Theory of Neural Computation. *Addison-Wesley Publishing Company*, CA, USA.

Hopfield, J.J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences*, vol. 79, pp. 2554—2558, USA.

Hornik, K., M. Stinchcombe and H. Whitw (1989). Multilayer feedforward networks. *Neural Networks*, vol. 2, pp. 359—366.

Kennedy, M. P. and L. O. Chua (1988). Neural networks for nonlinear programming. *IEEE Trans. On Circuits and Systems*, vol. 35, nº 5, pp. 554—562.

Khachiyan L. G. (1979). A Polynomial Algorithm in Linear Programming. *Soviet Mathematics Doklady*, vol. 20, pp. 191—194.

Kojima, M., S. Mizuno and A. Yoshise (1989). *A primal--dual interior point algorithm for linear programming, Progress in Mathematical Programming: Interior Point and Related Methods.* Springer Verlag, New York, pp. 29—47.

Ku, K. W. C., M. W. Mak and W.-C. Siu (2000). A study of the Lamarckian evolution of recurrent neural networks. *IEEE Trans. on Evolutionary Computation,* vol. 4, nº 1, pp. 31—42.

Lemmon, M. and P. T. Szymanski (1994). Interior Point Implementations of Alternating Minimization Training. *Advances in Neural Information Processing Systems*, vol. 7, pp. 574—582.

Lillo, W. E., M. H. Loh, S. Hui and H. S. Zak (1993). On Solving Constrained Optimization Problems with Neural Networks: A Penalty Method Approach. *IEEE Trans. Neural Networks,* vol. 4, nº 6, pp.931—940.

Luenberger, D. (1984). *Linear and Nonlinear Programming.* Addison-Wesley, Reading, Massachusetts.

Lustig, I. J., R. E. Marsten and D. F. Shanno (1992). On implementing Mehrotra's predictor-corrector interior-point method for linear programming. *SIAM J. Optimization*, vol. 2, nº 3, pp. 435—449.

Maa, C.–Y. and M. A. Shanblatt (1992a). Linear and Quadratic Programming Neural Network Analysis. *IEEE Trans. Neural Networks,* vol. 3, nº 4, pp. 580—594.

Maa, C.–Y. and M. A. Shanblatt (1992b). A Two-Phase Optimization Neural Network. *IEEE Trans. Neural Networks*, vol. 3, nº 6, pp. 1003—1009.

McCulloch, W.S. and Pitts, W. (1943). A Logical Calculus of the ideas immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115—133.

Mehrotra, S. (1992). On the Implementation of a Primal-Dual Interior Point Method. *SIAM Journal on Optimization*, vol. 2, nº 4, pp. 575—601.

Michalewicz, Z. (1992). *Genetic Algorithms +Data Structures = Evolution Programs*. Springer-Verlag, New York.

Michalewicz, Z. and D. B. Fogel (2000). *How to Solve It: Modern Heuristics*. Springer-Verlag, New York, USA.

Minsky, M.L. and S.A. Papert (1969). *Perceptrons*. MA:MIT Press*.,* Cambridge.

Monteiro, R. D. C., Ilan Adler and M. G. C. Resende (1990). A Polynomial-time Primal-Dual Affine Scaling Algorithm for Linear and Convex Quadratic Programming and its Power Series Extension. *Mathematics of Operations Research,* vol. 15, pp. 191—214.

Osowski, S. (1992). Neural Network for Non-Linear Programming with Linear Equality Constraints. *Int. J. of Circuit Theory and Applications,* vol. 20, pp. 93—98.

Pedrycz, W. and Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*. The MIT Press, Cambridge, Massachusetts.

Perez-Ilzarbe, M. J. (1998). Convergence Analysis of a Discrete-Time Recurrent Neural Network to Perform Quadratic Real Optimization with Bound Constraints. *IEEE Trans. on Neural Networks,* vol. 9, nº 6, pp. 1344—1351.

Reifman, J. and E. E. Feldman (1999). Nonlinear Programming with Feedforward Neural Networks. *IJCNN-International Joint Conference on Neural Networks*, vol.1, pp. 594—598.

Pham, D. T. and D. Karaboga (2000). *Intelligent Optimisation Techniques: genetic algorithms, tabu search, simulated annealing and neural networks*. Springer-Verlag London Limited.

Rodriguez-Vasquez, A., R. Dominguez-Castro, A. Rueda, J.L. Huertas and E. Sanchez-Sinnencio (1990). Nonlinear Switched-Capacitor 'Neural' Networks for Optimization Problems. *IEEE Trans. Circuits and Systems,* vol. **CAS-37**, nº 3, pp. 384—398.

Romero, R. A. F. (1993). *Otimização de Sistemas através de Redes Neurais Artificiais*. Tese de Doutorado, FEEC/UNICAMP.

Romero, R. A. F. (1996). Otimização de Sistemas através de Redes Neurais Multi-camadas. *XI-Congresso Brasileiro de Automática* vol. 2, pp. 1585—1590, São Paulo, Brasil.

Rooij, A. J. F., L. C. Jain and R. P. Johnson (1996). *Neural Network Training using Genetic Algorithms*. World Scientific Publishing Co. Pte. Ltd.

Roos, C., T. Terlaky and J.-PH. Vial (1997). *Theory and Algorithms for Linear Optimization*. John Wiley & Sons Ltd, England.

Rosenblatt, F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, vol. 65, pp. 386—408.

Saravanan, N. and D. B. Fogel (1995). Evolving Neural Control Systems. *Evolutionary Programming,* (June), pp. 23—27.

Silva, I. N. da (1997). Uma Abordagem Neuro-Nebulosa para Otimização de Sistemas e Identificação Robusta. Tese de Doutorado, FEEC/UNICAMP.

Silva, I. N. da, Lúcia V. Ramos de Arruda and Wagner Caradori do Amaral (1997). Robust estimation of parametric membership regions using artificial neural networks. *International Journal of Systems Science*, vol. 28, nº 5, pp. 447—455.

Silva, I. N. da, L. V. Ramos de Arruda and Wagner Caradori do Amaral (1998). "Nonlinear Optimization Using a Modified Hopfield Model", *IEEE-World Congress on Computational Intelligence*, pp. 1629—1633.

Silva, I. N. da, M. E. Bordon and A. N. de Souza (1999). Design and Analysis of Neural Networks for Systems Optimization, *IEEE International Joint Conference on Neural Networks*, pp. 684—689.

Szymanski, P. T., M. Lemmon and C. J. Bett (1998). Hybrid Interior Point Training of Modular Neural Networks. *Neural Networks*, vol. 11, pp. 215—234.

Tank, D. W. and J. J. Hopfield (1986). Simple 'Neural' Optimization Network: an A\D Converter, Signal Decision Circuit and a Linear Programming Circuit. *IEEE Trans. Circuits and Systems*, vol. CAS-33, pp. 533—541.

Tapia, R. A. and Yin Zhang (1992). Superlinear and Quadratic Convergence of Primal-Dual Interior Point Methods for Linear Programming Revisited. *Journal of Optimization Theory and Applications*, vol. 73, pp. 229—242.

Trafalis, T. B. and N. P. Couellan (1994). Neural Network Training via a Primal-Dual Interior Point Method for Linear Programming. *WCNN—World Congress on Neural Networks*, vol. 2, pp. 798—803.

Trafalis, T. B. and N. P. Couellan (1996). Neural Network Training via an Affine Scaling Quadratic Optimization Algorithm. *Neural Networks*, vol. 9, pp. 475—481.

Trafalis, T.B., N. P. Couellan and S. C. Bertrand (1997). Training of Supervised Neural Networks via a Nonlinear Primal-Dual Interior-Point Method. *IEEE-Proceedings of International Conference on Neural Networks*, pp. 2017—2021.

Tsuchiya, T. (1996). Chapter 2- Affine Scaling Algorithm. *Interior Point Methods of Mathematical Programming.* Kluwer Academic Publishers, Netherlands, pp. 35—82.

Velazco, M. I. e Christiano Lyra (2000). Otimização através de Redes Neurais Treinadas com Algoritmos Genéticos. *XIV-Congresso Brasileiro de Automática*, Florianópolis/SC, Brasil, pp. 229—234.

Velazco, M. I. and C. Lyra (2002). Optimization with Neural Networks Trained by Evolutionary Algorithms. *IJNN—International Joint Conference on Neural Netowks*, Honolulu, Hawaii/EUA, pp. 1516—1521.

Velazco, M. I., R. L. Oliveira and C. Lyra (2002a). Neural Networks Give a Warm Start to Linear Optimization. *IJNN—International Joint Conference on Neural Netowks*, vol. 2, pp. 1871—1876.

Velazco, M. I., R. L. Oliveira e C. Lyra (2002b). Inicialização Inteligente para Métodos de Pontos Interiores através de Redes Neurais de Hopfield. *XIV-Congresso Brasileiro de Automática*, Natal/RN, Brasil, pp. 27—31.

Velazco, M. I., R. L. Oliveira e C. Lyra (2003). Hopfield Neural Networks Flavor Large Optimization Problems. Submitted to *Neural Networks.*

Wright, S. J. (1996). *Primal--Dual Interior--Point Methods*. SIAM Publications, SIAM, Philadelphia, PA, USA.

Wu, AI and P. K. S. Tam (1999). Using Neural Network Method computes Quadratic Optimization Problems. *IEEE- Third International Conference on Computational Intelligence and Multimidia Applications*, (Sept.), pp. 70—74.

Xia, Y-S. and J. Wang (1995). Neural Networks for Solving Linear Programming Problems with Bounded Variables. *IEEE Trans. Neural Networks*, vol. 6, n^{o} 2, pp. 515—519.

Xia, Y-S. (1996). A New Neural Network for Solving Linear Programming Problems and its Application. *IEEE Trans. Neural Networks*, vol. 7, n^{o}. 2, pp. 525—529.

Xia Y-S. (1996a). A New Neural Network for Solving Linear and Quadratic Programming Problems. *IEEE Trans. Neural Networks*, vol. 7, nº 6, pp. 1544—1547.

Xia, Y-S. and J. Wang (1998). A General Methodology for Designing Globally Convergent Optimization Neural Networks. *IEEE Trans. Neural Networks,* vol. 9, nº 6, pp. 1331—1343.

Xia Y-S. and J. Wang (2000). Global Exponential Stability of Recurrent Neural Networks for Solving Optimization and Related Problems. *IEEE Trans. Neural Networks*, vol. 11, nº 4, pp. 1017—1022.

Yildirim, E. A. and S. J. Wright (2002). Warm-Start Strategies in Interior-Point Methods for Linear Programming. *SIAM J. Optimization*, vol. 12, nº 3, pp. 782—810.

Zhang, S., X. Zhu and L.-H. Zou (1992). Second-Order Neural Net for Constrained Optimization. *IEEE Trans. Neural Networks*, vol. 3, nº 6, pp. 1021—1024.

Zhang, Y. (1998). Solving Large-Scale Linear Programs by Interior-Point Methods Under The Matlab Environment. *Optimization Methods & Software*, vol. 10, nº 1, pp. 1—31.

APÊNDICE A.

OTIMIZAÇÃO POR REDES NEURAIS EM COOPERAÇÃO COM ALGORITMOS GENÉTICOS

As áreas de algoritmos genéticos e redes neurais podem ser encontradas trabalhando cooperativamente de várias formas [Michalewicz e Fogel 2000]; algoritmos genéticos utilizados para o treinamento de redes multicamadas [Ku *et al.*1995], algoritmos genéticos utilizados para gerar a arquitetura das redes com os pesos sendo definidos a partir de técnicas de otimização não-linear irrestrita [Iyoda *et al.* 1999] ou métodos evolutivos para gerar os pesos e arquitetura da rede [Rooij *et al.* 1996]. Entre as aplicações destes sistemas, podemos mencionar: sistemas de controle [Saravanan e Fogel 1995], previsões financeiras [Harrald e Kamstra 1997], aprendizado supervisionado [Castro *et al.* 1998] (para determinar boas condições iniciais).

Chellapilla e Fogel utilizaram estas técnicas para melhorar o desempenho de máquinas de jogos [Chellapilla e Fogel 1999; Chellapilla e Fogel 2000, Chellapilla e Fogel 1999]. Neste caso, redes neurais e algoritmos genéticos são utilizados para aprender estratégias para os jogos dilema do prisioneiro, tic-tac-toe e xadrez. No caso do jogo de xadrez, a rede neural gerada por computação evolutiva é capaz de jogar no nível de esperteza [Chellapilla e Fogel 2000].

Nesta seção, é apresentada uma forma diferente de cooperação destas técnicas, onde redes neurais e algoritmos genéticos solucionam problemas de otimização. Na nova abordagem, uma rede neural multicamada definida para resolver problemas de otimização [Romero 1996] utiliza algoritmos genéticos como método de otimização na atualização dos pesos. Esta abordagem, chamada de Neuro-Evolutiva, foi inicialmente apresentada em [Velazco e Lyra 2000] onde foi obtido um desempenho não tão bom quanto à abordagem apresentada em [Velazco e Lyra 2002].

As redes neurais multicamadas para otimização foram definidos por Romero em [Romero 1993, Romero 1996]. São redes de três camadas com igual número de neurônios em cada uma delas; na qual a função de erro quadrático, tradicionalmente utilizada para este tipo de rede, é substituída pela função objetivo do problema de otimização. O método do gradiente realiza a otimização da função através da atualização dos pesos.

A abordagem Neuro-Evolutiva substitui o método do gradiente por algoritmos genéticos para a atualização dos pesos. Esta metodologia tem a vantagem de não precisar de informação de primeira ordem da função objetivo para a solução do problema, o que permite trabalhar com problemas não diferenciáveis e não contínuos. Adicionalmente, os algoritmos genéticos permitem uma busca quase-global no espaço de solução, vantagem muito útil para problemas não-convexos.

O desempenho da abordagem Neuro-Evolutiva é medido na resolução de quatro problemas de otimização irrestrita: dois problemas convexos e dois não-convexos. Esses

resultados são comparados quanto ao número de iterações com algoritmos genéticos puros e com a abordagem de Romero, quando aplicados a os mesmos problemas.

A.1 COMPUTAÇÃO EVOLUTIVA E ALGORITMOS GENÉTICOS

Nesta seção, são apresentados conceitos gerais sobre algoritmos genéticos que permitirão a melhor compreensão deste trabalho.

Algoritmos genéticos são heurísticas de otimização baseadas na teoria da evolução. Nesses algoritmos genéticos, a partir de uma população inicial, ou conjunto de indivíduos, é realizada a reprodução através da aplicação dos operadores de mutação e cruzamento. Os indivíduos da próxima geração são escolhidos pelo operador de seleção.

Na aplicação de algoritmos genéticos para a solução de um problema, devem ser analisadas quatro componentes:

- A representação genética dos indivíduos ou variáveis do problema;
- A função de adequação do problema, que determina a adaptação ao meio;
- Os operadores de reprodução por mutação e cruzamento, que permitem a expansão da população;
- O operador de seleção, que determina quais indivíduos formarão a próxima geração.

Para abordar um problema através de algoritmos genéticos, o primeiro passo é procurar uma representação genética para o mesmo. O indivíduo é codificado através do cromossomo que o identifica na população. A representação matemática do vetor é realizada através de um vetor com componentes binárias ou reais. Um cromossomo binário é uma cadeia de elementos com valores no conjunto $\{0,1\}$ e um cromossomo com representação real é formado por elementos com valores no domínio dos reais.

A função de adequação ou *fitness* é utilizada para medir a capacidade de adaptação do indivíduo ao meio. Esta função é utilizada na seleção do indivíduo para a próxima geração.

A reprodução é realizada através dos operadores de mutação e cruzamento. Estes operadores são definidos de acordo com a codificação utilizada. Cada operador atua sobre um ou mais indivíduos da população. Com isto, novos indivíduos são obtidos, os quais formarão a população intermediária. Um operador de seleção [Michalewicz 1992] atua na população intermediária para definir a geração seguinte.

A.2 OTIMIZAÇÃO POR REDES NEURAIS MULTICAMADAS

As redes neurais multicamadas para otimização foram definidas por Romero [Romero 1993] para a solução de problemas de otimização convexa definidos como segue:

$$\begin{aligned} &\min_{x} f(x) \\ s.a \quad &r(x) = 0 \\ &g(x) \le 0 \\ &x \in \Psi \subseteq \Re^n \end{aligned} \tag{A.1}$$

onde $f : \Re^n \to \Re$, $r : \Re^n \to \Re^p$ e $g : \Re^n \to \Re^q$.

A rede é formada por três camadas de igual número de neurônios: uma camada de entrada, uma camada intermediária e uma camada de saída, como mostra a fig. A.1. Cada camada está totalmente interconectada com a camada seguinte e possui tantos neurônios quanto variáveis do problema de otimização. A saída da rede é retro-alimentada à camada de entrada. Cada neurônio, inclusive os da camada de saída, possuem uma função de ativação sigmoidal,

$$g(x) = \frac{1}{1 + e^x}, \tag{A.2}$$

que tem o espaço de solução definido na região:

$$\Omega = [0,1]. \tag{A.3}$$

Em uma rede neural multicamada, é otimizada uma função de erro quadrático entre a saída desejada e a saída obtida em cada iteração [Haykin 1999]. Esta otimização é feita através de um método de otimização que emprega retro-propagação do erro para atualizar os pesos em cada camada. Em redes neurais multicamadas para otimização, a função de erro quadrático é substituída por uma função de otimização irrestrita obtida a partir do problema original (Equação A.1). A otimização é realizada sobre os pesos definidos nas conexões entre neurônios através do método do gradiente, e é utilizada a regra de atualização empregada em redes neurais multicamadas. A saída da rede é retro-alimentada se as condições de convergências não forem atingidas; caso sejam atingidas, o ótimo foi encontrado.

A camada de saída utiliza a função de ativação (A.2) que tem um espaço de solução Ω. Isto restringe o conjunto de problemas a serem resolvidos pois todos têm que estar definidos na região Ω, incorporada como uma restrição de canalização para o problema (A.1).

A seleção da topologia da rede é justificada por Romero [1996] em três pontos:

1. Uma função pode ser aproximada por uma rede neural de três camadas [Funahashi 1989; Hornik *et al.* 1989];

2. A saída da rede é retro-alimentada na rede porque o método do gradiente é utilizado na atualização dos pesos e o problema de otimização é convexo pelo que se pode supor que a saída obtida é mais próxima da saída desejada. Com isto, a saída deve ser utilizada como próximo ponto na otimização;

3. A regra de atualização dos pesos pode ser obtida de forma análoga àquela empregada para uma rede neural multicamada.

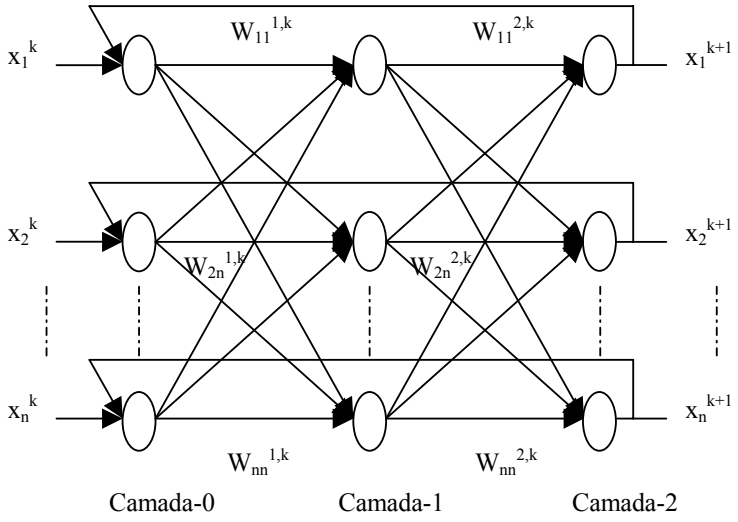

Fig. A.1. Rede Neural Multicamada para Otimização

A metodologia desta rede pode ser resumida como segue. Considere um ponto inicial x^0 (um vetor de componentes x_i^0, $x_i^0 \in [0,1]$, $i = 1,\ldots,n$).

I. Fazer até a convergência:

i. Propagar o ponto x^k pela rede;

ii. Verificar condição de parada a partir da saída obtida x_{out}^k; caso positivo ir passo II;

iii. Calcular a nova atualização dos pesos;

iv. Fazer $x^{k+1} = x_{out}^k$, voltar passo I.i.

II. Fazer $x^* = x_{out}^k$.

As redes multicamadas para otimização trabalham com problemas restritos e irrestritos, mas a otimização é realizada a partir de uma função de otimização irrestrita.

Os problemas de otimização com restrições são transformados em problemas irrestritos utilizando a teoria de dualidade [Romero 1993].

A regra de atualização dos pesos e a condição de convergência são definidas de forma diferente para ambos os tipos de problemas, mas neste trabalho só serão apresentadas estas definições para problemas irrestritos. Os problemas restritos não foram tratados na abordagem com algoritmos evolutivos.

A.2.1 Regra de Atualização dos Pesos

A regra de atualização dos pesos é obtida de forma análoga à definida para a atualização dos pesos em redes neurais multicamadas [Haykin 1999].

Em redes multicamadas, é definida uma função de erro quadrático entre a saída desejada (t) e a saída obtida (u).

$$E = \frac{1}{2}\sum_j \left(t_j - u_j\right)^2 \tag{A.4}$$

A função de erro quadrático na solução de problemas de otimização irrestrita, em redes neurais multicamadas para otimização, é substituída pela função objetivo do problema de otimização:

$$E(x) = f(x). \tag{A.5}$$

Ao aplicar a regra de atualização dos pesos de redes neurais multicamadas a esta nova definição da função de erro obtemos:

$$w_{ij}^{p+1} = w_{ij}^{p} + \Delta w_{ij}^{p}, \tag{A.6}$$

$$\Delta w_{ij} = -\frac{\partial E}{\partial w_{ij}} = -\frac{\partial f}{\partial w_{ij}}, \tag{A.7}$$

$$\Delta w_{ij}^{p} = \eta \cdot \delta_{j}^{p} \cdot u_{i}^{p-1}, \tag{A.8}$$

onde,

w_{ij}^{p} é o peso da conexão entre o neurônio *i* e o neurônio *j* na camada *p*;

η é o passo;

$\delta_{j}^{(p)}$ é definida nas equações (A.11) e (A.12) como a derivada parcial da função *E* com respeito à *j*-ésima componente da camada *p*, com sinal contrário;

$u_{i}^{(p-1)}$ é a *i*-ésima saída da camada *p-1*.

A saída de cada camada é definida como:

$$u_{i}^{(p)} = g\left(y_{i}^{p}\right), \tag{A.9}$$

$$y_{i}^{(p)} = \sum_{i} w_{ij}^{(p)} \cdot u_{i}^{(p-1)}. \tag{A.10}$$

A derivada de *E* com respeito à entrada é diferente em cada camada, como mostram as seguintes equações:

$$\delta_{j}^{(2)} = -\frac{\partial f}{\partial u_{j}^{(2)}}\left(u^{(2)}\right) \cdot g'\left(y_{j}^{(2)}\right), \tag{A.11}$$

$$\delta_{j}^{(1)} = \left(\sum_{i} \delta_{i}^{(2)} \cdot w_{ji}^{(2)}\right) \cdot g'\left(y_{j}^{(1)}\right). \tag{A.12}$$

O passo η na abordagem de Romero é fixo e definido previamente ao processo de convergência da rede. Este valor é calculado aleatoriamente no intervalo $[0,1]$.

Esta forma de definição do passo pode influenciar muito na convergência do método:

- Se o passo for muito pequeno, a convergência é muito lenta quando estamos longe do mínimo;
- Se o passo for muito grande, nas primeiras iterações longe do ótimo a convergência é rápida, mas quando estamos perto de um mínimo o método pode oscilar.

A.2.2 Condição de Parada

A condição de parada da rede é definida a partir do cálculo do jacobiano da função objetivo. Estas redes trabalham com problemas convexos pelo que, quando o Jacobiano é nulo, se pode afirmar que o mínimo local é o mínimo global do problema. Para verificar esta condição, utiliza-se a seguinte expressão:

$$\|\nabla f(x)\| \leq \varepsilon\,, \tag{A.13}$$

onde ε é uma valor no intervalo $(0,1)$ e $\|v\|$ é definido como:

$$\|v\| = \max_i |v_i|, \qquad v^T = [v_1, \ldots, v_n]. \tag{A.14}$$

A.3 A ABORDAGEM NEURO-EVOLUTIVA

A abordagem Neuro-Evolutiva foi definida utilizando as redes neurais multicamadas para otimização e algoritmos genéticos na atualização dos pesos da rede.

As modificações apresentadas na nova abordagem estão baseadas em três pontos:

1. A função de ativação sigmodal só permite trabalhar com problemas definidos no intervalo $[0,1]$ para todas as variáveis, pelo que uma nova função de ativação é utilizada. A nova função de ativação é a função rampa que permite aumentar o espaço de solução dos problemas;

2. Somente são utilizados problemas convexos; o método de otimização só garante convergência global nesses casos. A utilização dos algoritmos genéticos como método de otimização permite uma busca quase-global no espaço de soluções, pelo que podemos tratar problemas não-convexos.

3. O método do gradiente é utilizado como técnica de otimização, de modo que não é possível trabalhar com problemas não diferenciáveis e não contínuos. Os algoritmos genéticos não trabalham com informação de primeira ordem da função objetivo, de modo que uma maior variedade de problemas podem ser abordados.

Na abordagem Neuro-Evolutiva, a otimização é realizada através de algoritmos genéticos sobre os pesos da rede. A cada iteração, é obtido um novo ponto que é utilizado como nova entrada da rede na próxima geração se a função de adequação assim o determinar. Um indivíduo é composto por dois elementos: o cromossomo de pesos da rede e a sua entrada. Na geração *k,* uma população é formada por *l* indivíduos e sobre o cromossomo são aplicados os operadores de mutação e cruzamento. Como resultado, são obtidas *m* populações intermediárias. Os indivíduos desta população estão formados pelos novos pesos e pela entrada da rede herdada do pai. Os melhores indivíduos são escolhidos para a próxima geração *k+1*, isto é, o modelo elitista é o método utilizado na seleção [Michalewicz 1992]. O sistema pode ser interpretado como *n* redes neurais simultaneamente procurando o mínimo de uma função.

A.3.1 Indivíduo

Na abordagem Neuro-Evolutiva, o indivíduo está composto por dois elementos: o cromossomo de pesos da rede e o vetor de entrada. Para a representação do cromossomo de pesos foi utilizada a representação real. O cromossomo é um vetor de tamanho $2 \cdot n^2$ de número reais (*n* é o tamanho do problema). Cada gene ou elemento *i* do vetor é limitado no intervalo predefinido $[l_i, u_i]$ e os operadores são obrigados a cumprir esse requerimento.

O vetor de entrada que representa a variável *x* do problema utiliza a codificação real; nenhuma operação é realizada sobre ele.

A.3.2 Função de Adequação

Seja *p* um indivíduo da população intermediária; obtido depois da aplicação dos operadores de reprodução sobre o cromossomo de pesos do pai. O indivíduo *p* está formado pelo cromossomo de pesos *w* e o vetor de entrada *x* herdado do pai,

$$p = (x, w). \tag{A.15}$$

O cálculo do valor da função de adequação *h* é realizado em três passos:

1. O vetor de entrada herdado do pai é propagado na rede com o novo cromossomo de pesos *w*; um novo vetor de saída $\tilde{x}$ é obtido, isto é,

$$rede(x, w) = \tilde{x} \tag{A.16}$$

2. A função *h* é calculada como o mínimo, ou máximo (dependendo do problema), entre o valor da função objetivo avaliada no vetor de entrada da rede *x* e o vetor de saída $\tilde{x}$; isto é,

$$h(p) = min(f(x), f(\tilde{x})), \tag{A.17}$$

se o problema for de minimização, ou

$$h(p) = max(f(x), f(\tilde{x})), \tag{A.18}$$

se o problema for de maximização;

3. Se o valor da função de adequação for obtido a partir do vetor de saída, então este vetor é colocado como vetor de entrada no indivíduo (Equação A.19). Se o valor da função de adequação é obtido a partir do vetor de entrada, então este continua como vetor de entrada do indivíduo (Equação A.20).

$$p = (\tilde{x}, w) \qquad se \quad h(p) = f(\tilde{x}), \tag{A.19}$$

$$p = (x, w) \qquad se \quad h(p) = f(x) \tag{A.20}$$

O novo indivíduo *p* será escolhido para a próxima geração se o seu valor na função de adequação pertence ao conjunto dos *m* indivíduos melhor adaptados. Quando o problema de otimização é de minimização, a próxima geração é composta pelos indivíduos com os menores valores da função *h*; em problemas de maximização, adoto-se procedimento simétrico.

A.3.3 Operadores

Para a obtenção dos indivíduos da população intermediária são aplicados os operadores de mutação e cruzamento sobre o cromossomo de pesos da rede. Esta seção apresenta os operadores de mutação e cruzamento que foram utilizados na reprodução [Michalewicz 1992, Herrera *et al.* 1994].

O objetivo é obter a maior variedade de novos indivíduos que permitia uma melhor busca no espaço de solução do problema, pelo que foram utilizados dez operadores de mutação e cruzamento.

Os operadores são aplicados sobre o cromossomo de pesos e definidos para o trabalho com representação em ponto flutuante. A seleção dos cromossomos para a reprodução é realizada aleatoriamente ou pelo método roleta russa de acordo com a exigência do operador. Os genes de cada cromossomo são restringidos no intervalo $[l_i, u_i]$.

A. Operadores de Mutação

Foram definidos três operadores de mutação.

1. **Mutação com deslocamento**. Os genes selecionados aleatoriamente são modificados a partir da suma o resta de um valor aleatório, ε. Este valor é obtido a

partir de um parâmetro de entrada, γ que determina o deslocamento máximo, $\varepsilon \in [-\gamma, \gamma]$ [Herrera *et al.* 1994].

2. **Mutação Aleatória**. Os genes selecionados são modificados por valores aleatórios com distribuição uniforme no domínio de definição, $[l_i, u_i]$ [Michalewicz 1992]. Este operador permite aumentar a diversidade populacional.

3. **Mutação não-uniforme**. Os genes selecionados g_i são substituídos por novos genes $\tilde{g}_i$ definidos como:

$$\tilde{g}_i = \begin{cases} g_i + \Delta(k, u_i - g_i) & if \quad a = 0 \\ g_i + \Delta(k, g_i - l_i) & if \quad a = 1 \end{cases}, \tag{A.21}$$

onde o valor de *a* é selecionado aleatoriamente do conjunto $\{0,1\}$, *p* é a geração atual e a função $\Delta(k, x)$ retorna um valor no intervalo $[0, x]$, tal que a probabilidade de que o seu valor seja perto de *0* aumenta proporcionalmente ao aumento da geração *k* [Michalewicz 1992]. Quando *k* é pequeno (nas gerações inicias) este operador permite obter cromossomos mais afastados do pai, mas com o aumento das gerações são obtidos pontos mais próximos ao pai—são realizadas buscas locais.

A. Operadores de Cruzamento

Foram definidos sete operadores de cruzamento. Para a geração de um novo indivíduo pelo cruzamento são envolvidos dois cromossomos; sejam $C' = (g'_1, g'_2, \ldots, g'_n)$ e $C'' = (g''_1, g''_2, \ldots, g''_n)$ os cromossomos selecionados.

1. **Cruzamento Plano.** O resultado é um cromossomo $\tilde{C} = (\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_n)$ no qual $\tilde{g}_i$ é um valor aleatório (distribuição uniforme) no intervalo $[g'_i, g''_i]$.

2. **Cruzamento Simples**. Uma posição $i \in \{1,2,\ldots,n\}$ em um cromossomo é selecionada aleatoriamente e dois novos cromossomos são obtidos $\widetilde{C} = (g'_1,\ldots,g'_i,g''_{i+1},\ldots,g''_n)$ e $\hat{C} = (g''_1,\ldots,g''_i,g'_{i+1},\ldots,g'_n)$ [Michalewicz 1992].

3. **Cruzamento Aritmético Uniforme.** Dois cromossomos $\widetilde{C} = (\tilde{g}_1,\ldots,\tilde{g}_n)$ e $\hat{C} = (\hat{g}_1,\ldots,\hat{g}_n)$ são gerados, onde $\tilde{g}_i = a \cdot g'_i + (1-a) \cdot g''_i$ e $\hat{g}_i = a \cdot g''_i + (1-a) \cdot g'_i$. O parâmetro *a* é uma constante previamente definida à evolução do algoritmo no intervalo $[0,1]$ [Michalewicz 1992].

4. **Cruzamento Aritmético Não-uniforme** [Michalewicz 1992]. Dois cromossomos $\widetilde{C} = (\tilde{g}_1,\ldots,\tilde{g}_n)$ e $\hat{C} = (\hat{g}_1,\ldots,\hat{g}_n)$ são gerados, onde $\tilde{g}_i = a \cdot g'_i + (1-a) \cdot g''_i$ e $\hat{g}_i = a \cdot g''_i + (1-a) \cdot g'_i$. O parâmetro *a* é um valor que decresce inversamente proporcional ao aumento do número de gerações:

$$a = \frac{1}{k} \tag{A.22}$$

onde *k* é a geração atual.

5. **Cruzamento BLX-α**. Gera um novo cromossomo $\widetilde{C} = (\tilde{g}_1,\ldots,\tilde{g}_n)$, onde $\tilde{g}_i$ é um valor aleatório com distribuição uniforme no intervalo $[x_i - I \cdot \alpha \cdot y_i + I \cdot \alpha]$, $y_i = max(g'_i, g''_i)$, $x_i = min(g'_i, g''_i)$ e $I = y_i - x_i$.

6. **Cruzamento Discreto**. Gera um novo cromossomo $\widetilde{C} = (\tilde{g}_1,\ldots,\tilde{g}_n)$, onde $\tilde{g}_i$ escolhido aleatoriamente do conjunto $\{g'_i, g''_2\}$.

7. **Cruzamento Intermediário Estendido**. Gera um novo cromossomo $\widetilde{C} = (\tilde{g}_1,\ldots,\tilde{g}_n)$, onde os genes são definidos como $\tilde{g}_i = g'_i + \alpha_i \cdot (g''_i - g'_i)$. O parâmetro α_i é um valor aleatório em um intervalo predefinido.

A.4 ESTUDO DE CASOS

Esta seção apresenta os resultados obtidos na aplicação da nova abordagem em quatro problemas de otimização irrestrita. O desempenho da abordagem Neuro-Evolutiva é comparado com o desempenho da metodologia de Romero e com a de algoritmos genéticos puros.

Os quatro problemas apresentados são problemas com solução conhecida. Foram utilizados dois problemas convexos e dois problemas não-convexos. Para cada abordagem foram rodados quatro testes com diferentes pontos iniciais calculados aleatoriamente em um intervalo especifico para cada abordagem. A inicialização dos pesos é feita também aleatoriamente no intervalo $[-2,2]$.

A.4.1 Aspectos de Implementação da Metodologia de Romero

A metodologia de Romero foi implementada com uma modificação. Romero utiliza funções de ativação sigmoidal nos neurônios de cada camada, isto restringe os problemas que podem ser otimizados; só podem ser utilizados problemas definidos no intervalo $[0,1]$. Para este trabalho foi utilizada uma função de ativação rampa permitindo um domínio de definição mais amplo.

Nos quatro testes realizados para cada problema foram utilizados passos, η, diferentes. O ponto inicial em cada teste foi calculado aleatoriamente no intervalo $[-100,100]$.

A.4.2 Aspectos de Implementação dos Algoritmos Genéticos Puros e da Abordagem Neuro-Evolutiva

A definição dos algoritmos genéticos puros foi realizada da seguinte forma:

- **Indivíduo:** vetor em ponto flutuante do tamanho do problema de otimização.
- **Função de adequação:** própria função objetivo do problema.

- **Operadores de reprodução.** utilizados para gerar a população intermediária foram os mesmos utilizados na abordagem Neuro-Evolutiva, definidos na Seção A.3.

A população inicial utilizada na abordagem Neuro-Evolutiva e na abordagem por algoritmos genéticos puros foi obtida aleatoriamente no intervalo $[-1000,1000]$.

Os valores dos parâmetros nos operadores de reprodução são definidos da mesma forma em ambas abordagens.

A.4.3 Problemas Convexos

Para apresentação dos resultados só serão considerados problemas irrestritos, pelo que a convexidade da função objetivo garante a convexidade do problema de otimização.

O primeiro problema convexo considerado é o problema quadrático de minimização representado na Equação (A.23). O mínimo global do problema é no ponto $x^* = (1,2)$.

$$min\, f(x) = (x_1 - 1)^2 + (x_2 - 2)^2, \qquad (x_1, x_2) \in \Re^2 \tag{A.23}$$

A função objetivo *f*,do problema é apresenta na Fig. A.2.

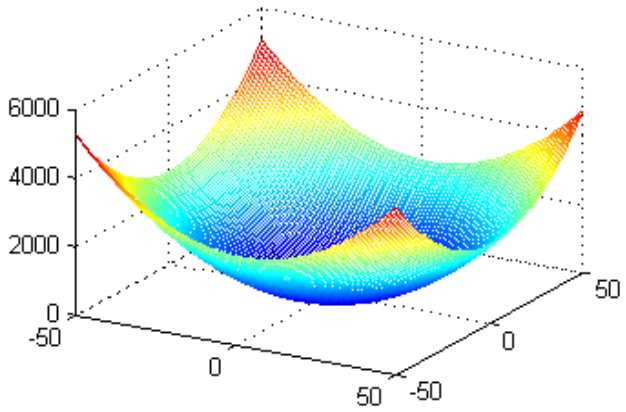

Fig. A.2. Função $f(x)=(x_1-1)^2 + (x_2-2)^2$.

A Tabela A.1 apresenta os resultados obtidos por cada abordagem nos quatro testes realizados com o problema da Equação A.23. Todas as abordagens obtiveram o ótimo, mas a abordagem Neuro-Evolutiva conseguiu em média em menor número de iterações.

TABELA A.1. RESULTADOS OBTIDOS COM O PROBLEMA $min\ f(x)$

ABORDAGEM	TESTE 1	TESTE 2	TESTE 3	TESTE 4
ROMERO	*15*	*20*	*20*	*22*
NEURO-EVOLUTIVA	*5*	*10*	*15*	*19*
ALGORITMOS GENÉTICOS	*12*	*18*	*19*	*20*

O segundo problema de minimização é:

$$min\ g(x) = e^{x_1 - x_2} + e^{x_2 - x_1} + e^{x_1^2} + x_3^2 - 3 \qquad (x_1, x_2, x_3) \in \Re^3 \tag{A.24}$$

O mínimo global deste problema é no ponto $x = (0,0,0)$.

A Tabela A.2 mostra os resultados obtidos pelas três abordagens utilizadas. Todas alcançaram o ótimo do problema, mas a abordagem Neuro-Evolutiva chegou em menor número de iterações.

TABELA A.2. RESULTADOS OBTIDOS COM O PROBLEMA $min\ g(x)$

ABORDAGENS	TESTE 1	TESTE 2	TESTE 3	TESTE 4
ROMERO	22	25	30	31
NEURO-EVOLUTIVA	2	2	2	2
ALGORITMOS GENÉTICOS	703	800	946	988

A.4.4 Problemas Não-Convexos

Foram considerados dois problemas não-convexos sobre o $\Re^2$.

O primeiro problema é:

$$min\ h(x) = x_1^3 + x_2^3 - 3 \cdot x_1 - 12 \cdot x_2 + 20, \qquad (x_1, x_2) \in \Re^2 \tag{A.25}$$

Esta função tem um mínimo local no ponto $x = (1,2)$, mas é ilimitada, como mostra a Fig. A.3.

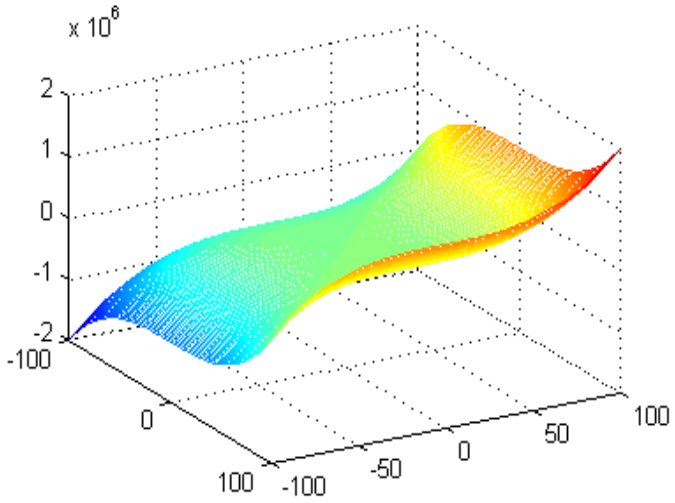

Fig. A.3. Função $h(x) = x_1^3 + x_2^3 - 3 \cdot x_1 - 12 \cdot x_2 + 20$.

Os resultados dos testes realizados são apresentados na Tabela A.3. Como discutido anteriormente, a abordagem proposta por Romero trabalha com problemas convexos, pelo que o resultado obtido era o esperado; foi obtido o mínimo local do problema de otimização.

As abordagens Neuro-Evolutiva e algoritmos genéticos puros deram resultados coerentes com o comportamento do problema; foram obtidos valores de *x* no limite inferior imposto na implementação dos algoritmos.

TABELA A.3. RESULTADOS OBTIDOS COM O PROBLEMA $min\ h(x)$

ABORDAGENS	TESTE 1	TESTE 2	TESTE 3	TESTE 4
ROMERO	MÍNIMO LOCAL	MÍNIMO LOCAL	MÍNIMO LOCAL	MÍNIMO LOCAL
NEURO-EVOLUTIVA	-INFINITO	-INFINITO	-INFINITO	-INFINITO
ALGORITMOS GENÉTICOS	-INFINITO	-INFINITO	-INFINITO	-INFINITO

O segundo problema não-convexo é:

$$min \quad l(x) = 10 \cdot \left(sin\left(x_1^2 + x_2^2\right)\right)^2 + \sqrt{x_1^2 + x_2^2}, \qquad (x_1, x_2) \in \Re^2 \tag{A.26}$$

Este problema é considerado difícil pela ocorrência de muitos mínimos locais como mostra um corte da função no ponto $x = (0, x_2)$ da Fig. A.4. Esta função possui um mínimo global no ponto $x^* = (0,0)$.

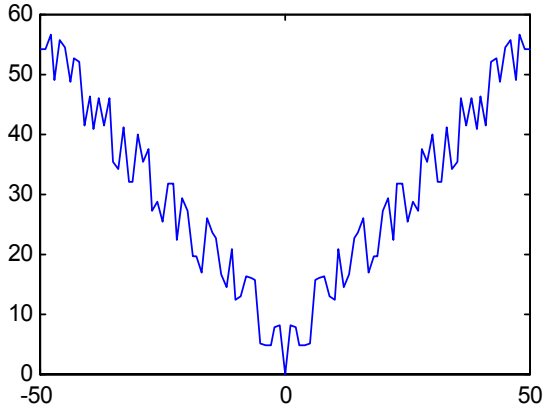

Figura A.4. Função $l(x) = 10 \cdot \left(sin\left(x_1^2 + x_2^2\right)\right)^2 + \sqrt{x_1^2 + x_2^2}$ no ponto $x = (0, x_2)$.

Os resultados dos testes realizados são apresentados na Tabela A.4.

A abordagem de Romero utiliza um passo fixo durante todo o processo de convergência, isto pode provocar as oscilações do método em um ponto de mínimo do

problema, especificamente quando se trabalha com este tipo de função com muitos mínimos locais e cheia de picos.

TABELA A.4. RESULTADOS OBTIDOS COM O PROBLEMA $min\, l(x)$

ABORDAGENS	**TESTE 1**	**TESTE 2**	**TESTE 3**	**TESTE 4**
ROMERO	MÉTODO OSCILA	MÉTODO OSCILA	MÍNIMO LOCAL	MÍNIMO LOCAL
NEURO-EVOLUTIVA	2	2	2	2
ALGORITMOS GENÉTICOS	22	26	27	28

As abordagens que utilizam algoritmos genéticos chegaram ao mínimo global do problema, mas a abordagem Neuro-Evolutiva chegou ao ótimo em número menor de iterações em todos os testes realizado.

A.5 RESUMO

Nesta seção, foi apresentada uma abordagem para solução de problemas de otimização através de redes neurais multicamadas e algoritmos genéticos. A abordagem chamada de Neuro-Evolutiva substitui a função de erro quadrático das redes multicamadas tradicionais pela função objetivo do problema e utiliza algoritmos genéticos na atualização dos pesos.

A nova abordagem foi testada em quatro problemas de otimização irrestrita. O desempenho da abordagem neuro-evolutiva é comparado com a abordagem de Romero e com uma abordagem por algoritmos genéticos puros levando em consideração o número de iterações. Na comparação, a abordagem proposta teve melhor desempenho em todos os problemas considerados.

APÊNDICE B.

ARTIGOS ASSOCIADOS AO TRABALHO

A pesquisa realizada nesta tese foi divulgada (até o momento) nos trabalhos listados a seguir:

M. I. Velazco e Christiano Lyra (2000). Otimização através de Redes Neurais Treinadas com Algoritmos Genéticos. *XIV-Congresso Brasileiro de Automática*, Florianópolis/SC, Brasil, pp. 229—234.

M. I. Velazco and C. Lyra (2002). Optimization with Neural Networks Trained by Evolutionary Algorithms. *IJNN—International Joint Conference on Neural Netwoks*, Honolulu, Hawaii/EUA, pp. 1516—1521.

M. I. Velazco, R. L. Oliveira and C. Lyra (2002a). Neural Networks Give a Warm Start to Linear Optimization. *IJNN—International Joint Conference on Neural Netwoks*, vol. 2, pp. 1871—1876.

M. I. Velazco, R. L. Oliveira e C. Lyra (2002b). Inicialização Inteligente para Métodos de Pontos Interiores através de Redes Neurais de Hopfield. *XIV-Congresso Brasileiro de Automática*, Natal/RN, Brasil, pp. 27—31.

M. I. Velazco, R. L. Oliveira e C. Lyra (2003). Hopfield Neural Networks Flavor Large Optimization Problems. Submitted to *Neural Networks.*